



Manual do usuário

Amazon ElastiCache



Versão da API 2015-02-02

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon ElastiCache: Manual do usuário

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é ElastiCache?	1
Armazenamento em cache sem servidor	1
Clusters baseados em nós	2
Serviços relacionados	2
Como funciona	3
Cache e mecanismos de cache	3
Escolha entre opções de implantação	9
ElastiCache recursos	16
AWSRegiões e zonas de disponibilidade da	18
Casos de uso	20
Começando com ElastiCache	30
Conf ElastiCacheiguração	30
Inscreva-se para um Conta da AWS	30
Criar um usuário com acesso administrativo	31
Conceder acesso programático	32
Configurar permissões	34
Configurar EC2	35
Conceder acesso de rede	36
Configurar o acesso à linha de comando	37
Criar um cache de tecnologia sem servidor do Valkey	38
Ler e gravar dados	39
Limpeza	41
Próximas etapas	42
Criar um cache de tecnologia sem servidor Redis OSS	42
Ler e gravar dados	43
Limpeza	45
Próximas etapas	46
Criar um cache de tecnologia sem servidor do Memcached	46
Ler e gravar dados	48
Limpeza	53
Próximas etapas	54
Tutoriais: conceitos básicos do Python e do ElastiCache	54
Python e ElastiCache	55
Tutorial: Configurando o Lambda para ElastiCache acesso em uma VPC	73

Etapa 1: Criar um cache ElastiCache sem servidor	73
Etapa 2: criar uma função Lambda para ElastiCache	76
Etapa 3: Teste a função Lambda com ElastiCache	80
Etapa 4: limpar (opcional)	80
Criação e gerenciamento de um cluster baseado em nós ElastiCache	82
Componentes e recursos	82
Nodes	83
ElastiCache cacos	84
ElastiCache aglomerados	84
ElastiCache replicação	86
ElastiCache endpoints	89
Grupos de parâmetros	90
ElastiCache segurança	90
Grupos de sub-redes	90
ElastiCache backups	91
Eventos	91
ElastiCache terminologia	92
Tutorial: Como criar um cluster baseado em nós ElastiCache	94
Criação de um ElastiCache cluster baseado em nós para o Valkey	94
Criação de um ElastiCache cluster baseado em nós para o Redis OSS	115
Excluir um cluster	137
Outros tutoriais e vídeos	139
Vídeos	140
Gerenciamento de nós no ElastiCache	145
Visualizando o status ElastiCache do nó	146
Nódulos e fragmentos do Valkey ou Redis OSS	151
Conexão a nós	154
Tipos de nó compatíveis	159
Reinicialização de nós	175
Substituição de nós (Valkey e Redis OSS)	180
Substituição de nós (Memcached)	187
Nós reservados	188
Migração de nós da geração anterior	205
Gerenciamento de clusters no ElastiCache	208
Escolhendo um tipo de rede em ElastiCache	211
Descoberta automática (Memcached)	216

Classificação de dados em níveis no ElastiCache	259
Preparação de um cluster no ElastiCache	266
Criação de um cluster do Valkey ou Redis OSS	277
Criação de um cluster do Memcached	287
Visualização dos detalhes de um cluster do ElastiCache	291
Modificação de um cluster do ElastiCache	305
Adicionar nós a um ElastiCache cluster	312
Removendo nós de um ElastiCache cluster	323
Cancelamento de operações de adição ou exclusão de nós pendentes no ElastiCache	332
Excluindo um cluster no ElastiCache	333
Acessando seu ElastiCache cluster ou grupo de replicação	336
Encontrar endpoints de conexão no ElastiCache	345
Fragmentos no ElastiCache	361
Comparação de clusters Valkey, Memcached e Redis OSS baseados em nós	367
Migração on-line para o Valkey ou Redis OSS	373
Visão geral	373
Etapas da migração	374
Preparação da origem e destino para migração	374
Testar a migração de dados	376
Início da migração	377
Verificação do andamento do processo de migração de dados	379
Conclusão da migração de dados	379
Execução de migração de dados online usando o console	380
Seleção de regiões e zonas de disponibilidade para o ElastiCache	382
Considerações sobre zonas de disponibilidade com o Memcached	383
Localização dos seus nós	385
Regiões e endpoints com suporte	385
Uso de zonas locais com o ElastiCache	391
Usando Outposts com ElastiCache	392
Trabalhando com ElastiCache	397
Snapshots e restauração	397
Restrições	398
Impacto sobre o desempenho dos backups de clusters baseados em nós	399
Programação de backups automáticos	401
Realização de backups manuais	403
Criação de um backup final	410

Descrição de backups	413
Copiar backups	415
Exportação de um backup	417
Restauração de um backup	425
Exclusão de um backup	426
Marcação de backups	428
Tutorial: propagação de um cluster baseado em nós com um backup	429
Versões e atualizações do motor em ElastiCache	439
Atualização de versões de mecanismos	439
Suporte estendido	444
Gerenciamento de versões para ElastiCache	448
Vale 8.2	455
Vale 8.1	455
Vale 8.0	456
ElastiCache versão 7.2.6 para Valkey	456
Versões compatíveis do Redis OSS	457
Cronograma de fim de vida das versões do Redis OSS	471
Versões do Memcached com suporte	473
Principais diferenças de comportamento e compatibilidade do mecanismo da versão com o Valkey	478
Principais diferenças de comportamento e compatibilidade do mecanismo da versão com o Redis OSS	478
Considerações sobre atualização ao trabalhar com clusters baseados em nós	483
Práticas recomendadas e estratégias de armazenamento em cache	486
Práticas recomendadas gerais	486
Práticas recomendadas para usar réplicas de leitura	487
Comandos Valkey, Memcached e Redis OSS compatíveis e restritos	491
Configuração e limites do Valkey e Redis OSS	529
Exemplos de clientes IPv6 para Valkey, Memcached e Redis OSS	533
Práticas recomendadas para clientes (Valkey e Redis OSS)	534
Práticas recomendadas para clientes (Memcached)	558
Clusters do ElastiCache de pilha dupla habilitados para TLS	562
Gerenciamento de memória reservada para Valkey e Redis OSS	566
Práticas recomendadas ao trabalhar com clusters baseados em nós do Valkey e Redis OSS	573
Estratégias de armazenamento em cache para Memcached	579

Gerenciando seu cluster baseado em nós no ElastiCache	585
Auto Scaling de clusters Valkey e Redis OSS	585
Modificar o modo de cluster	634
Replicação entre AWS regiões usando datastores globais	637
Alta disponibilidade com o uso de grupos de replicação	666
Gerenciamento de manutenção de cluster do ElastiCache	757
Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache	760
Conectar uma instância do EC2 e um cache do ElastiCache automaticamente	877
Conectividade EC2 automática	878
Visualizar recursos computacionais conectados	884
Dimensionamento ElastiCache	885
Dimensionamento sem servidor ElastiCache	885
Definição dos limites de escalabilidade para gerenciar custos	885
Pré-escalonamento com Serverless ElastiCache	885
Definindo limites de escala usando o console e AWS CLI	887
Dimensionamento de clusters baseados em nós	888
Conceitos básicos do filtro Bloom	975
Tipo de dados Bloom	976
Limite de tamanho do bloom	977
ACLs	977
Métricas relacionadas ao Bloom	978
Comandos de filtro Bloom	979
Introdução ao Watch em Serverless	980
Limite do relógio	981
Comandos suportados relacionados ao Watch	981
Conceitos básicos de pesquisa vetorial	982
Visão geral sobre a pesquisa vetorial	982
Atributos e limites da pesquisa vetorial	987
Escolher a configuração apropriada	989
Comandos de pesquisa vetorial	990
Conceitos básicos do JSON para Valkey e Redis OSS	1003
Visão geral do tipo de dados do JSON	1004
Comandos JSON	1016
Marcando seus recursos ElastiCache	1059
Monitoramento de custos com tags	1072

Gerenciar tags usando o AWS CLI	1073
Gerenciamento de tags usando a API do ElastiCache	1078
Lente do Well-Architected para o Amazon ElastiCache	1081
Pilar Excelência operacional	1082
Pilar Segurança	1091
Pilar Confiabilidade	1097
Pilar Eficiência de performance	1103
Pilar Otimização de custos	1114
Solução de problemas em ElastiCache	1121
Problemas de conexão	1121
Erros do cliente Valkey ou Redis OSS	1122
Solução de problemas de alta latência no Serverless ElastiCache	1123
Solução de problemas de limitação no Serverless ElastiCache	1125
Problemas de conexão persistentes	1126
Related Topics	1147
Segurança	1148
Proteção de dados	1149
Segurança de dados no Amazon ElastiCache	1149
Privacidade do tráfego entre redes	1228
Amazon VPCs e ElastiCache segurança	1229
ElastiCache API e interface VPC endpoints ()AWS PrivateLink	1254
Sub-redes e grupos de sub-redes	1260
Gerenciamento de Identidade e Acesso	1269
Público	1269
Autenticação com identidades	1269
Gerenciar o acesso usando políticas	1271
Como a Amazon ElastiCache trabalha com o IAM	1273
Exemplos de políticas baseadas em identidade	1278
Solução de problemas	1281
Controle de acesso	1283
Visão geral do gerenciamento de acesso	1284
Validação de conformidade	1335
Mais informações	1336
Resiliência	1336
Atenuar falhas	1337
Segurança da infraestrutura	1342

Atualizações de serviço	1342
Gerenciamento das atualizações de serviços	1343
Vulnerabilidades de segurança tratadas	1348
Registro em log e monitoramento	1351
Métricas e eventos para Valkey e Redis OSS com tecnologia sem servidor	1351
Métricas sem servidor	1351
Eventos sem servidor	1362
Métricas e eventos de cluster baseados em nós para Valkey e Redis OSS	1376
Métricas para cluster baseados em nós	1376
Eventos para cluster baseado em nós (Valkey e Redis OSS)	1376
Métricas e eventos para o Memcached	1385
Métricas com tecnologia sem servidor do Memcached	1385
Eventos com tecnologia sem servidor do Memcached	1389
Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail	1399
Informações sobre o Amazon ElastiCache no CloudTrail	1400
Noções básicas sobre entradas de arquivo de log do Amazon ElastiCache	1401
Monitoramento de eventos do Amazon SNS	1404
Gerenciando ElastiCache notificações do Amazon SNS	1405
Visualizando ElastiCache eventos	1410
Notificações de eventos e o Amazon SNS	1414
Entrega de logs	1422
Conteúdo de uma entrada de log lento	1423
Conteúdo de uma entrada de log do mecanismo	1423
Permissões para configurar os logs	1423
Especificações de tipo de log e formato de log	1424
ElastiCache destinos de registro	1425
Especificação da entrega de logs usando o console	1428
Especificação da entrega de log usando a AWS CLI	1429
Monitoramento do uso	1435
Métricas em nível de host	1435
Métricas para o Valkey e Redis OSS	1439
Métricas para o Memcached	1458
Que métricas devo monitorar?	1464
Escolher estatísticas e períodos de métricas	1469
Monitorando métricas de CloudWatch clusters e nós	1469
Cotas	1473

Referência	1476
Uso da API do ElastiCache	1476
Como usar a API de consulta	1476
Bibliotecas disponíveis	1480
Solução de problemas de aplicações	1480
Configurar a AWS CLI para o ElastiCache	1481
Pré-requisitos	1482
Obtenção das ferramentas da linha de comando	1484
Configuração das ferramentas	1484
Como fornecer credenciais para as ferramentas	1485
Variáveis de ambiente	1486
Mensagens de erro	1487
Notificações	1489
Notificações gerais do ElastiCache	1489
Notificações do ElastiCache for Memcached	1489
Notificações específicas do ElastiCache para Redis OSS	1490
ElastiCache Histórico da documentação	1491
AWS Glossário	1532
.....	mdxxxiii

O que é a Amazon ElastiCache?

Bem-vindo ao Guia do ElastiCache usuário da Amazon. ElastiCache A Amazon é um serviço web que facilita a configuração, o gerenciamento e a escalabilidade de um armazenamento de dados distribuído na memória ou ambiente de cache na nuvem. Ele fornece uma solução de armazenamento em cache econômica, de alta performance e escalável. Ao mesmo tempo, ele ajuda a remover a complexidade associada à implantação e ao gerenciamento de um ambiente de cache distribuído.

Você pode operar a Amazon ElastiCache em dois formatos. Você pode começar com um cache com tecnologia sem servidor ou criar um cluster baseado em nós.

Note

A Amazon ElastiCache trabalha com os mecanismos OSS Valkey, Memcached e Redis. Se não souber qual mecanismo deseja usar, consulte [Comparação de clusters Valkey, Memcached e Redis OSS baseados em nós](#) neste guia.

Armazenamento em cache sem servidor

ElastiCache oferece armazenamento em cache sem servidor, o que simplifica a adição e a operação de um cache para seu aplicativo. ElastiCache O Serverless permite que você crie um cache altamente disponível em menos de um minuto e elimina a necessidade de provisionar instâncias ou configurar nós ou clusters. Os desenvolvedores podem criar um cache sem servidor especificando o nome do cache usando o ElastiCache console, o SDK ou a CLI.

ElastiCache A tecnologia sem servidor também elimina a necessidade de planejar e gerenciar a capacidade de armazenamento em cache. ElastiCache monitora constantemente a memória, a computação e a largura de banda de rede do cache usadas pelo seu aplicativo e é dimensionado para atender às necessidades do seu aplicativo. ElastiCache oferece uma experiência de endpoint simples para desenvolvedores, abstraindo a infraestrutura de cache subjacente e o design do cluster. ElastiCache gerencia o provisionamento de hardware, o monitoramento, a substituição de nós e a aplicação de patches de software de forma automática e transparente, para que você possa se concentrar no desenvolvimento de aplicativos, em vez de operar o cache.

ElastiCache O Serverless é compatível com Valkey 7.2, Memcached 1.6.21 e superior e Redis OSS 7.1 e superior.

Criação de um cluster baseado em nós

Se você precisar de um controle refinado sobre seu ElastiCache cluster, poderá optar por criar um cluster OSS Valkey, Memcached ou Redis baseado em nós. ElastiCache permite criar um cluster baseado em nós escolhendo o tipo de nó, o número de nós e o posicionamento dos nós nas zonas de AWS disponibilidade do seu cluster. Como ElastiCache é um serviço totalmente gerenciado, ele gerencia automaticamente o provisionamento de hardware, o monitoramento, as substituições de nós e a aplicação de patches de software para seu cluster.

Criar um cluster baseado em nós oferece mais flexibilidade e controle sobre os clusters. Por exemplo, você pode optar por operar um cluster com disponibilidade single-AZ ou disponibilidade multi-AZ, dependendo das necessidades. Você também pode optar por executar o Valkey, Memcached ou Redis OSS em modo de cluster habilitando a escalabilidade horizontal, ou sem o modo de cluster, para escalar apenas verticalmente. Ao criar um cluster baseado em nós, você é responsável por escolher o tipo e o número de nós corretamente para garantir que o cache tenha capacidade suficiente, conforme exigido pela aplicação. Você também pode escolher quando aplicar novos patches de software ao cluster do Valkey ou Redis OSS.

Ao criar um cluster baseado em nós, você pode escolher entre várias versões compatíveis do Valkey, Memcached e Redis OSS. Para obter mais informações sobre as versões de engine compatíveis, consulte [Versões e atualizações do motor em ElastiCache](#).

Serviços relacionados

[MemoryDB](#)

Ao decidir se deseja usar o ElastiCache ou o MemoryDB, considere as seguintes comparações:

- O ElastiCache é um serviço comumente usado para armazenar em cache dados de outros bancos de dados e armazenamento de dados usando o Valkey, Memcached ou Redis OSS. Você deve considerar o ElastiCache para armazenar workloads em cache nas quais você deseja acelerar o acesso aos dados com seu banco de dados principal ou armazenamento de dados existente (desempenho de leitura e gravação em microssegundos). Você também deve considerar o ElastiCache para casos de uso em que você deseja usar as estruturas de dados e APIs do Valkey ou Redis OSS para acessar dados armazenados em um banco de dados ou armazenamento de dados primário.
- O ElastiCache também pode ajudar você a economizar custos de banco de dados armazenando os dados mais acessados em um cache. Se a aplicação tiver requisitos de throughput de leitura

elevados, você poderá obter alta escala, desempenho rápido e custos de armazenamento de dados reduzidos usando o ElastiCache, em vez de escalar o banco de dados subjacente.

- O MemoryDB é um banco de dados em memória, durável para workloads que exigem um banco de dados primário ultrarrápido. É compatível com Valkey e Redis OSS. Você deve considerar o uso do MemoryDB se sua workload exigir um banco de dados durável que ofereça desempenho ultrarrápido (leitura de microssegundos e latência de gravação em menos de 10 milissegundos). O MemoryDB também pode ser uma boa opção para seu caso de uso se você quiser criar uma aplicação usando estruturas de dados e APIs do Valkey ou Redis OSS com um banco de dados primário e durável. Finalmente, você deve considerar o uso do MemoryDB para simplificar a arquitetura da aplicação e reduzir os custos substituindo o uso de um banco de dados por um cache para maior durabilidade e desempenho.

[Amazon Relational Database Service](#)

O ElastiCache pode ajudar você a economizar em custos de banco de dados armazenando os dados mais acessados em um cache. Se a aplicação tiver requisitos de throughput de leitura elevados, você poderá obter alta escala, desempenho rápido e custos de armazenamento de dados reduzidos usando o ElastiCache, em vez de escalar o banco de dados subjacente.

Para obter mais informações básicas sobre o serviço Amazon Relational Database Service relacionado, consulte [Amazon RDS](#)

O ElastiCache pode ajudar você a economizar em custos de banco de dados armazenando os dados mais acessados em um cache. Se a aplicação tiver requisitos de throughput de leitura elevados, você poderá obter alta escala, desempenho rápido e custos de armazenamento de dados reduzidos usando o ElastiCache, em vez de escalar o banco de dados subjacente.

Como ElastiCache funciona

Aqui você encontra uma visão geral dos principais componentes de uma ElastiCache implantação.

Cache e mecanismos de cache

Um cache é um armazenamento de dados na memória que é possível usar para armazenar dados em cache. Normalmente, seu aplicativo armazena em cache os dados acessados com frequência em um cache para otimizar os tempos de resposta. ElastiCache oferece duas opções de implantação: caches sem servidor e clusters baseados em nós. Consulte [Escolha entre opções de implantação](#).

 Note

A Amazon ElastiCache trabalha com os mecanismos OSS Valkey, Memcached e Redis. Se não souber qual mecanismo deseja usar, consulte [Comparação de clusters Valkey, Memcached e Redis OSS baseados em nós](#) neste guia.

Tópicos

- [Como ElastiCache funciona](#)
- [Dimensões de preço](#)
- [ElastiCache backups](#)

Como ElastiCache funciona

ElastiCache Sem servidor

ElastiCache O Serverless permite que você crie um cache sem se preocupar com planejamento de capacidade, gerenciamento de hardware ou design de cluster. Basta fornecer um nome para o cache, e você receberá um único endpoint que pode ser configurado no cliente do Valkey, Memcached, Redis OSS para começar a acessar o cache.

 Note

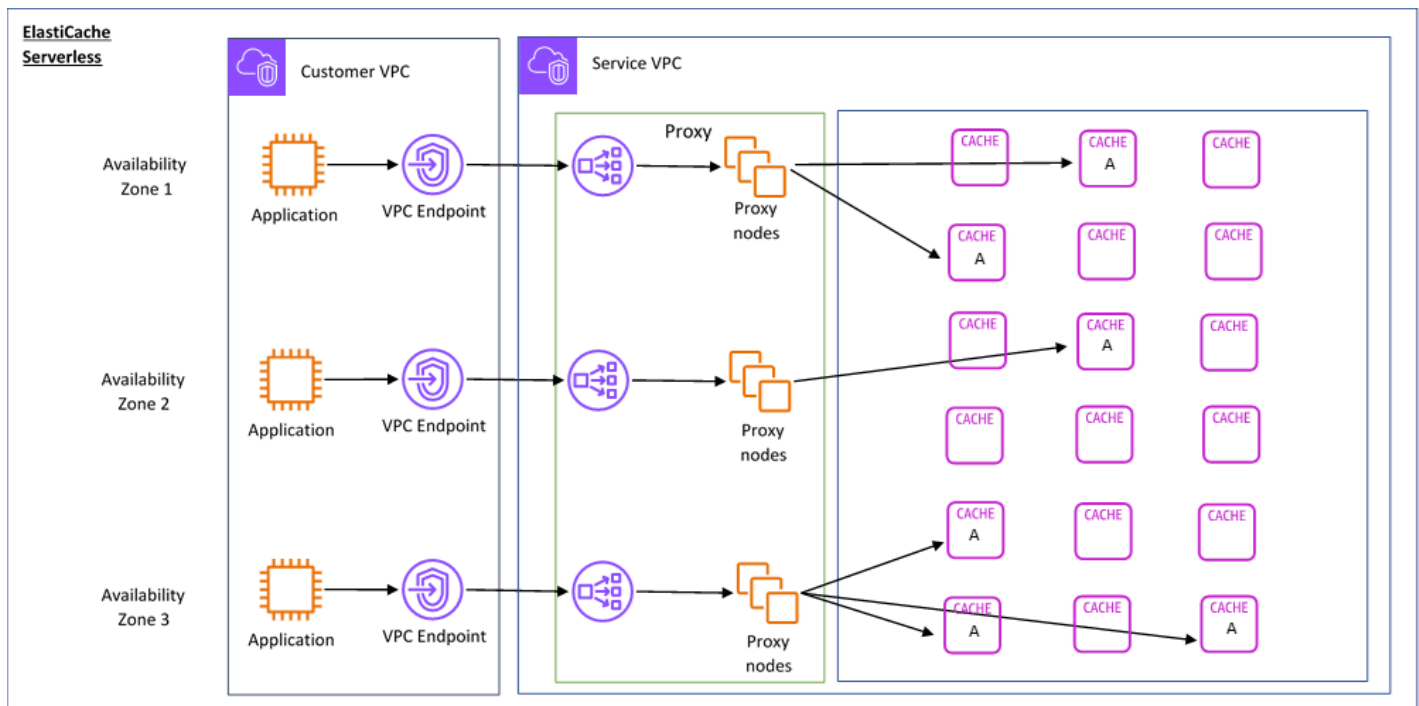
- ElastiCache O Serverless executa Valkey, Memcached ou Redis OSS no modo de cluster e só é compatível com clientes que oferecem suporte a TLS.

Benefícios principais

- Sem planejamento de capacidade: o ElastiCache Serverless elimina a necessidade de planejar a capacidade. ElastiCache O Serverless monitora continuamente a utilização de memória, computação e largura de banda de rede do seu cache e escala vertical e horizontalmente. Ele permite que um nó de cache cresça e, paralelamente, inicia uma operação de aumento de escala horizontal para garantir que o cache possa escalar para atender aos requisitos da aplicação em todos os momentos.

- **Pay-per-use:** Com o ElastiCache Serverless, você paga pelos dados armazenados e pela computação utilizada pela sua carga de trabalho no cache. Consulte [Dimensões de preço](#).
- **Alta disponibilidade:** o ElastiCache Serverless replica automaticamente seus dados em várias zonas de disponibilidade (AZ) para obter alta disponibilidade. Ele monitora automaticamente os nós de cache subjacentes e os substitui em caso de falhas. Ele oferece um SLA de disponibilidade de 99,99% para cada cache.
- **Atualizações automáticas de software:** o ElastiCache Serverless atualiza automaticamente seu cache para a última versão secundária e corrige o software sem nenhum impacto na disponibilidade do seu aplicativo. Quando uma nova versão principal estiver disponível, você ElastiCache receberá uma notificação.
- **Segurança:** o ElastiCache Sem Servidor sempre criptografa os dados em trânsito e em repouso. É possível usar uma chave gerenciada pelo serviço ou usar a própria chave gerenciada pelo cliente para criptografar dados em repouso.

O diagrama a seguir ilustra como o ElastiCache Serverless funciona.



Quando você cria um novo cache sem servidor, ElastiCache cria um Endpoint de Nuvem Privada Virtual (VPC) nas sub-redes de sua escolha em sua VPC. A aplicação pode se conectar ao cache por meio desses endpoints da VPC.

Com o ElastiCache Serverless, você recebe um único endpoint DNS ao qual seu aplicativo se conecta. Quando você solicita uma nova conexão com o endpoint, o ElastiCache Serverless manipula todas as conexões de cache por meio de uma camada de proxy. A camada proxy ajuda a reduzir a configuração do cliente complexa, porque o cliente não precisa redescobrir a topologia do cluster em caso de alterações no cluster subjacente. A camada proxy é um conjunto de nós proxy que processam conexões usando um balanceador de carga da rede.

Quando a aplicação cria uma nova conexão de cache, a solicitação é enviada para um nó proxy pelo balanceador de carga da rede. Quando a aplicação executa comandos de cache, o nó proxy conectado à aplicação executa as solicitações em um nó de cache no cache. A camada proxy abstrai a topologia do cluster e os nós do cliente. Isso permite balancear ElastiCache a carga de forma inteligente, escalar e adicionar novos nós de cache, substituir os nós de cache quando eles falham e atualizar o software nos nós de cache, tudo sem afetar a disponibilidade do seu aplicativo ou a necessidade de redefinir as conexões.

Clusters baseados em nós

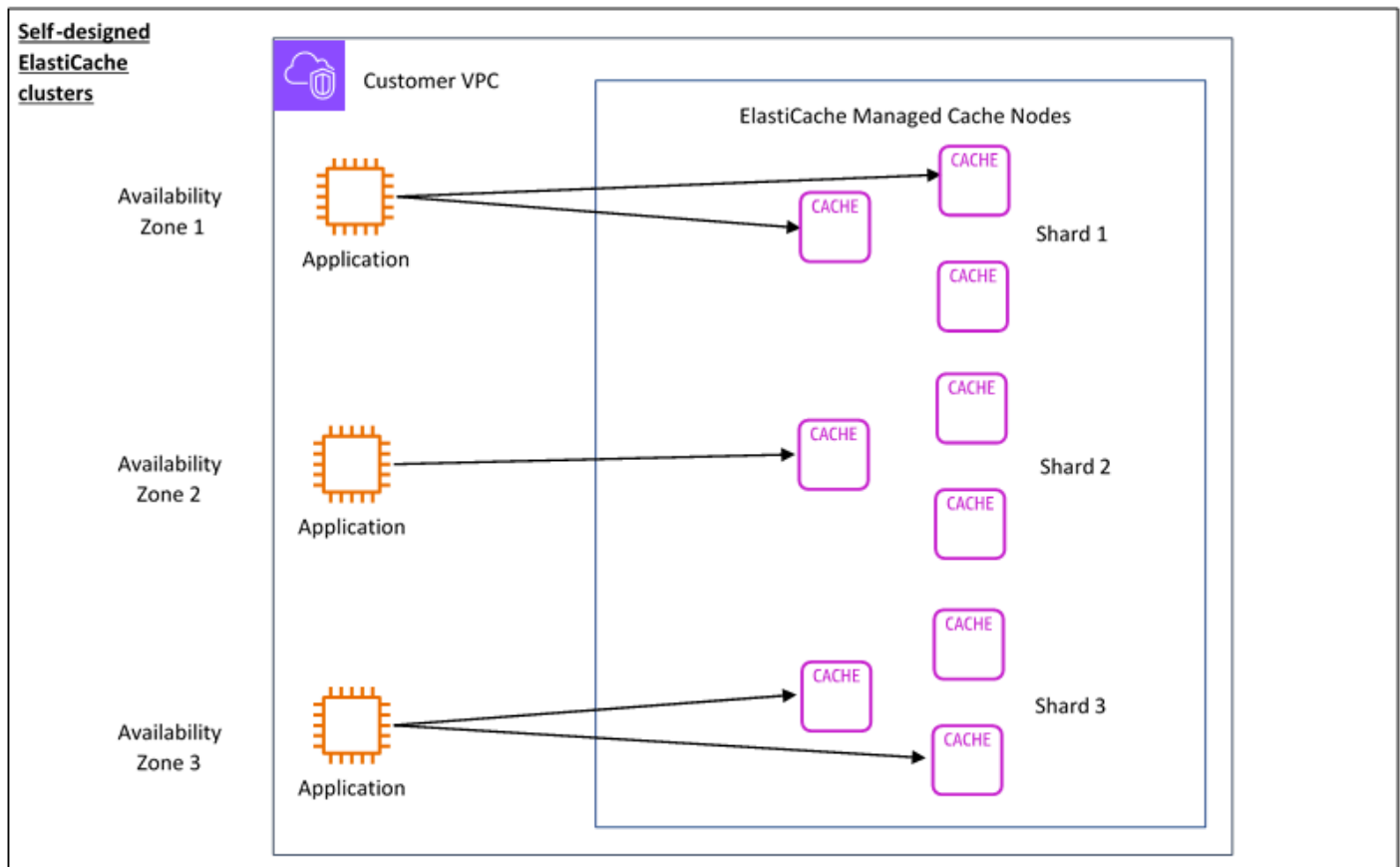
Você pode criar um ElastiCache cluster baseado em nós escolhendo uma família de nós de cache, tamanho e número de nós para seu cluster. Criar um cluster baseado em nós proporciona um controle mais refinado e permite a você escolher o número de fragmentos no cache e o número de nós (primário e réplica) em cada fragmento. Você pode optar por operar o Valkey ou Redis OSS em modo de cluster criando um cluster com vários fragmentos ou em modo sem cluster com um único fragmento.

Benefícios principais

- Crie um cluster baseado em nós: com ElastiCache, você pode criar um cluster baseado em nós e escolher onde deseja colocar seus nós de cache. Por exemplo, se tiver uma aplicação que deseja trocar a alta disponibilidade pela baixa latência, você poderá optar por implantar os nós de cache em uma única AZ. Como alternativa, você pode criar um cluster baseado em nós com nós em vários AZs para obter alta disponibilidade.
- Controle refinado: ao criar um cluster baseado em nós, você tem mais controle sobre o ajuste fino das configurações no cache. Por exemplo, é possível usar [Parâmetros do Valkey e do Redis OSS](#) or [Parâmetros específicos do Memcached](#) para configurar o mecanismo de cache.
- Escalar vertical e horizontalmente: você pode optar por escalar manualmente o cluster aumentando ou diminuindo o tamanho do nó de cache quando necessário. Você também pode escalar horizontalmente ao adicionar novos fragmentos ou adicionar mais réplicas aos fragmentos. Também é possível usar o recurso ajuste de escala automático para configurar o escalonamento

com base em uma programação ou o escalonamento com base em métricas como uso de CPU e memória no cache.

O diagrama a seguir ilustra como os clusters baseados em nós ElastiCache funcionam.



Dimensões de preço

Você pode implantar ElastiCache em duas opções de implantação. Ao implantar o ElastiCache Serverless, você paga pelo uso dos dados armazenados em GB/hora e pela computação em ElastiCache unidades de processamento (ECPU). Ao criar um cluster baseado em nós, você paga por hora de uso do nó de cache. Consulte detalhes de preço [aqui](#).

Armazenamento de dados

Você paga pelos dados armazenados no ElastiCache Serverless cobrados em gigabytes por hora (GB-hrs). ElastiCache O Serverless monitora continuamente os dados armazenados em seu cache, fazendo amostragens várias vezes por minuto, e calcula uma média horária para determinar o uso do

armazenamento de dados do cache em GB-h. Cada cache ElastiCache sem servidor é medido para um mínimo de 1 GB de dados armazenados.

ElastiCache Unidades de processamento (ECPUs)

Você paga pelas solicitações que seu aplicativo executa no ElastiCache Serverless in ElastiCache Processing Units (ECPUs), uma unidade que inclui o tempo de vCPU e os dados transferidos.

- Leituras e gravações simples exigem um ECPU para cada quilobyte (KB) de dados transferidos. Por exemplo, um comando GET que transfere até 1 KB de dados consome uma ECPU. Uma solicitação SET que transfere 3,2 KB de dados consumirá 3,2 ECPUs.
- Com o Valkey e o Redis OSS, os comandos que consomem mais tempo de vCPU e transferem mais dados consomem ECPUs com base na maior das duas dimensões. Por exemplo, se seu aplicativo usa o comando HMGET, consome 3 vezes mais tempo de vCPU do que um SET/GET comando simples e transfere 3,2 KB de dados, ele consumirá 3,2 ECPU. Como alternativa, se ele transferir apenas 2 KB de dados, consumirá 3 ECPUs.
- Com o Valkey e o Redis OSS, os comandos que exigem tempo adicional de vCPU consumirão proporcionalmente mais. ECPUs Por exemplo, se seu aplicativo usa o comando Valkey ou Redis OSS [HMGET e consome 3 vezes o tempo da vCPU como um comando](#) SET/GET simples, ele consumirá 3. ECPUs
- Com o Memcached, os comandos que operam em vários itens consumirão proporcionalmente mais. ECPUs Por exemplo, se seu aplicativo executar um multiget em 3 itens, ele consumirá 3 ECPUs.
- Com o Memcached, os comandos que operam em mais itens e transferem mais dados consomem ECPUs com base na maior das duas dimensões. Por exemplo, se a aplicação usar o comando GET, recuperar três itens e transferir 3,2 KB de dados, ela consumirá 3,2 ECPU. Como alternativa, se ele transferir apenas 2 KB de dados, consumirá 3 ECPUs.

ElastiCache O Serverless emite uma nova métrica chamada `ElastiCacheProcessingUnits` que ajuda você a entender o que é ECPUs consumido pela sua carga de trabalho.

Horas de nó

Você pode criar um cluster baseado em nós escolhendo a família de EC2 nós, o tamanho, o número de nós e o posicionamento nas zonas de disponibilidade. Ao criar um cluster baseado em nós, você paga por hora para cada nó de cache.

ElastiCache backups

Um backup é uma point-in-time cópia de um cache sem servidor ou de um cluster baseado em nós Valkey ou Redis OSS. ElastiCache permite que você faça um backup de seus dados a qualquer momento ou configure backups automáticos. Backups podem ser usados para restaurar um cache existente ou para propagar um novo cache. Backups consistem em todos os dados de um cache mais alguns metadados. Para obter mais informações, consulte [Snapshots e restauração](#).

Escolha entre opções de implantação

A Amazon ElastiCache tem duas opções de implantação:

- Armazenamento em cache sem servidor
- Clusters baseados em nós

Para obter uma lista de comandos aceitos para ambos, consulte [Comandos Valkey, Memcached e Redis OSS compatíveis e restritos](#).

Armazenamento em cache sem servidor

O Amazon ElastiCache Serverless simplifica a criação de cache e escala instantaneamente para dar suporte aos aplicativos mais exigentes dos clientes. Com o ElastiCache Serverless, você pode criar um cache altamente disponível e escalável em menos de um minuto, eliminando a necessidade de provisionar, planejar e gerenciar a capacidade do cluster. ElastiCache O Serverless armazena automaticamente os dados de forma redundante em três zonas de disponibilidade e fornece um Acordo de Nível de Serviço (SLA) de 99,99% de disponibilidade. Os backups de clusters Valkey baseados em nós ou de Redis OSS podem ser restaurados em uma configuração com tecnologia sem servidor.

Clusters baseados em nós

Se você precisar de um controle refinado sobre seu cluster OSS Valkey, Memcached ou Redis, você pode criar um cluster baseado em nós com ElastiCache. Você escolhe o tipo de nó, o número de nós e o posicionamento dos nós nas zonas de AWS disponibilidade do seu cluster. Como ElastiCache é um serviço totalmente gerenciado, ele ajuda a gerenciar o provisionamento de hardware, o monitoramento, as substituições de nós e a aplicação de patches de software para seu cluster. Clusters baseados em nós podem ser projetados para fornecer um SLA de disponibilidade de até 99,99%. Backups de caches Valkey ou Redis OSS de tecnologia sem servidor podem ser restaurados em um cluster baseado em nós.

Escolha entre opções de implantação

Escolha o armazenamento sem servidor se:

- Você está criando um cache para workloads novas ou difíceis de prever.
- Você tem tráfego de aplicativos imprevisível.
- Você deseja a maneira mais fácil de começar a usar um cache.

Crie seu próprio cluster baseado em nós se:

- Você já está executando o ElastiCache Serverless e deseja um controle mais refinado sobre o tipo de nó que executa o Valkey, o Memcached ou o Redis OSS, o número de nós e o posicionamento desses nós.
- Você espera que o tráfego do seu aplicativo seja relativamente previsível e deseja um controle refinado sobre desempenho, disponibilidade e custo.
- Você pode prever os requisitos de capacidade para controlar os custos.

Comparar cache com tecnologia sem servidor e clusters baseados em nós

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
Configuração de cache	Crie um cache com apenas um nome em menos de um minuto	Fornece controle refinado sobre o design do cluster. O usuário pode escolher o tipo de nó, o número de nós e o posicionamento nas zonas AWS de disponibilidade
ElastiCache Versão suportada	Valkey 7.2 e superior, Redis OSS versão 7.1 e superior, Memcached 1.6.21 e superior	Valkey 7.2 e superior, Redis OSS versão 4.0 e superior, Memcached 1.4 e superior
Modo Cluster (Valkey e Redis OSS)	Opera motores somente em <code>cluster mode enabled</code> . Os clientes devem oferecer suporte <code>cluster mode</code>	Pode ser configurado para operar no modo cluster habilitado ou desabilitado.

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
	enabled para se conectar ao ElastiCache Serverless.	
Escalabilidade	Escala automaticamente os mecanismos vertical e horizontalmente sem qualquer gerenciamento de capacidade.	Fornecer controle sobre o escalonamento, ao mesmo tempo em que exige monitoramento para garantir que a capacidade atual esteja atendendo adequadamente à demanda. Para Valkey e Redis OSS, você pode escolher escalar verticalmente aumentando ou diminuindo o tamanho do nó de cache quando necessário. Você também pode escalar horizontalmente ao adicionar novos fragmentos ou adicionar mais réplicas aos fragmentos. Esse recurso não está disponível para o Memcached. Com o atributo de ajuste de escala automático, você também pode configurar o dimensionamento com base em uma programação ou em métricas como uso de CPU e memória no cache.

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
Conexão do cliente	Os clientes se conectam a um único endpoint. Isso permite que a topologia do nó de cache subjacente (escalonamento, substituições e atualizações) mude sem desconectar o cliente.	Os clientes se conectam a cada nó de cache individual. Se um nó for substituído, o cliente redescobre a topologia do cluster e restabelece as conexões.
Configure	Nenhuma configuração refinada disponível. Os clientes podem configurar configurações básicas, incluindo sub-redes que podem acessar o cache, se os backups automáticos estão ativados ou desativados e limites máximos de uso do cache.	Clusters baseados em nós oferecem opções de configuração refinadas. Os clientes podem usar grupos de parâmetros para um controle refinado. Para uma tabela desses valores de parâmetro por tipo de nó, consulte Parâmetros específicos do mecanismo .
multi-AZ	Os dados são replicados de forma assíncrona em várias Zonas de Disponibilidade para maior disponibilidade e melhor latência de leitura.	Fornece a opção de criar o cluster em uma única zona de disponibilidade ou em várias zonas de disponibilidade (AZs). Ao usar o Valkey ou o Redis OSS, são fornecidos clusters Multi-AZ com dados replicados de forma assíncrona em várias zonas de disponibilidade para maior disponibilidade e melhor latência de leitura.

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
Criptografia em repouso	Sempre habilitado. Os clientes podem usar uma chave Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente AWS KMS.	Opção para habilitar ou desabilitar a criptografia em repouso. Quando ativada, os clientes podem usar uma chave Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente AWS KMS.
Criptografia em trânsito (TLS)	Sempre habilitado. Os clientes devem oferecer suporte à conectividade TLS.	Opção para habilitar ou desabilitar.
Backups	Aceita backups automáticos e manuais de caches sem impacto no desempenho. Os backups do Valkey e do Redis OSS são compatíveis entre si e podem ser restaurados em um cache ElastiCache sem servidor ou em um cluster baseado em nós.	Aceita backups automáticos e manuais para Valkey e Redis OSS. Os clusters podem ter algum impacto sobre o desempenho, dependendo da memória reservada disponível. Para obter mais informações, consulte Gerenciamento de memória reservada para Valkey e Redis OSS. Os backups do Valkey e do Redis OSS são compatíveis entre si e podem ser restaurados em um cache ElastiCache sem servidor ou em um cluster baseado em nós.

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
Monitoramento	Support métricas em nível de cache, incluindo taxa de acerto do cache, taxa de erro do cache, tamanho dos dados e ECPUs consumo. ElastiCache O Serverless envia eventos usando EventBridge quando eventos significativos acontecem em seu cache. Você pode escolher monitorar, ingerir, transformar e agir em ElastiCache eventos usando a Amazon EventBridge. Para obter mais informações, consulte Eventos de cache sem servidor.	ElastiCache Os clusters baseados em nós emitem métricas em cada nível de nó, incluindo métricas em nível de host e métricas de cache. Clusters baseados em nós emitem notificações de SNS para eventos significativos. Consulte Métricas para o Memcached e Métricas para o Valkey e Redis OSS.
Disponibilidade	Acordo de Serviço (SLA) com 99,99% de disponibilidade	Clusters baseados em nós podem ser projetados para atingir um Acordo de Serviço (SLA) de disponibilidade de até 99,99%, dependendo da configuração.

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
Atualizações e correções de software	Atualiza automaticamente o software de cache para a versão secundária e de patch mais recente, sem impacto no aplicativo. Os clientes recebem uma notificação sobre atualizações de versões principais e podem atualizar para a versão principal mais recente quando quiserem.	Os clusters baseados em nós oferecem autoatendimento habilitado pelo cliente para atualizações de versões secundárias e de patches, bem como atualizações de versões principais. As atualizações gerenciadas são aplicadas automaticamente durante as janelas de manutenção definidas pelo cliente. Os clientes também podem optar por aplicar um upgrade de versão secundária ou de patch sob demanda.
Global Data Store	Não compatível	Suporta Global Data Store, que permite replicação entre regiões com gravações de região única e leituras multirregionais

Recurso	Armazenamento em cache sem servidor	Clusters baseados em nós
Hierarquização de dados	Não compatível	Clusters que são criados usando nós da família r6gd têm seus dados classificados em níveis entre a memória e o armazenamento local em unidades de estado sólido (Solid state Drives, SSD). O armazenamento de dados em camadas fornece uma opção de preço-desempenho para cargas de trabalho Valkey e Redis OSS, utilizando unidades de estado sólido (SSDs) de baixo custo em cada nó do cluster, além de armazenar dados na memória.
Modelo de definição de preços	Pay-per-use, com base em dados armazenados em GB-hora e solicitações em Unidades de ElastiCache Processamento (ECPU). Consulte detalhes de preço aqui .	Pay-per-hour, com base no uso do nó de cache. Consulte detalhes de preço aqui .

Tópicos relacionados:

- [Criação e gerenciamento de um cluster baseado em nós ElastiCache](#)

ElastiCache Recursos da Amazon para usuários iniciantes

Recomendamos que os usuários iniciantes comecem lendo as seguintes seções e as consultem conforme necessário.

- Destaques e preços do serviço — A [página de detalhes do](#) produto fornece uma visão geral do produto ElastiCache, destaques do serviço e preços.
- ElastiCache vídeos — A [Vídeos do ElastiCache](#) seção tem vídeos que apresentam você à Amazon ElastiCache. Os vídeos abordam casos de uso comuns ElastiCache e demonstram como usar ElastiCache para reduzir a latência e melhorar a produtividade de seus aplicativos.
- Conceitos básicos a seção [Começando com a Amazon ElastiCache](#) inclui informações sobre a criação de um cache. Ele também inclui como autorizar o acesso ao cache, conectar-se a um nó de cache e excluir o cache.
- Desempenho em escala — O ElastiCache whitepaper [Performance at Scale with Amazon](#) aborda estratégias de armazenamento em cache que ajudam seu aplicativo a ter um bom desempenho em grande escala.

Depois de completar as seções anteriores, leia estas seções:

- [Escolha do tamanho do nó](#)

Você quer que seus nós sejam grandes o suficiente para acomodar todos os dados que deseja armazenar em cache. Ao mesmo tempo, você não quer pagar mais cache do que precisa. Você pode usar esse tópico para ajudar a selecionar o melhor tamanho do nó.

- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)

Identifique e aborde problemas que possam afetar a eficiência do seu cluster.

Se quiser usar o AWS Command Line Interface(AWS CLI), você pode usar esses documentos para ajudá-lo a começar:

- [AWS Command Line Interface documentação](#)

Esta seção fornece informações sobre como baixar o AWS CLI, fazer com que AWS CLI ele funcione em seu sistema e forneça suas AWS credenciais.

- [AWS CLI documentação para ElastiCache](#)

Esse documento separado abrange todos os ElastiCache comandos AWS CLI for, incluindo sintaxe e exemplos.

Você pode escrever programas de aplicativos para usar a ElastiCache API com uma variedade de linguagens de programação populares. Aqui estão alguns recursos:

- [Ferramentas para a Amazon Web Services](#)

A Amazon Web Services fornece vários kits de desenvolvimento de software (SDKs) com suporte para ElastiCache. Você pode programar para ElastiCache usar Java, .NET, PHP, Ruby e outras linguagens. Isso SDKs pode simplificar muito o desenvolvimento de seu aplicativo formatando suas solicitações ElastiCache, analisando respostas e fornecendo lógica de repetição e tratamento de erros.

- [Uso da API do ElastiCache](#)

Se você não quiser usar o AWS SDKs, você pode interagir ElastiCache diretamente usando a API de consulta. É possível encontrar dicas de solução de problemas e informações sobre como criar e autenticar solicitações e lidar com respostas nesta seção.

- [Referência de ElastiCache API da Amazon](#)

Esse documento separado abrange todas as operações da ElastiCache API, incluindo sintaxe e exemplos.

Regiões e zonas de disponibilidade da AWS

Os recursos de computação em nuvem da Amazon estão alojados em instalações de datacenter altamente disponíveis em diferentes áreas do mundo (por exemplo, América do Norte, Europa ou Ásia). Cada localização de datacenter é chamada de uma região da AWS.

Cada região da AWS contém vários locais distintos, chamados de zonas de disponibilidade, ou AZs. Cada zona de disponibilidade é projetada para ser isolada de falhas em outras zonas de disponibilidade. Cada uma é projetada para fornecer conectividade de rede de baixa latência e custo reduzido para outras zonas de disponibilidade na mesma região da AWS. Ao iniciar as instâncias em Zonas de disponibilidade separadas, você pode proteger seus aplicativos de falha de um único local. Para obter mais informações, consulte [Escolha de regiões e zonas de disponibilidade](#).

Você pode criar seu cluster em várias zonas de disponibilidade, uma opção chamada de implantação multi-AZ. Quando você escolhe essa opção, a Amazon automaticamente provisiona e mantém uma instância de nó secundário em espera em uma zona de disponibilidade diferente. A instância do seu nó primário é sincronicamente replicada ao longo das zonas de disponibilidade para a instância secundária. Essa abordagem ajuda a fornecer redundância de dados e suporte a failover, eliminar

congelamentos de E/S e minimizar picos de latência durante backups do sistema. Para obter mais informações, consulte [Minimização do tempo de inatividade no ElastiCache para Valkey e Redis OSS com o Multi-AZ](#).

Casos de ElastiCache uso comuns e como ElastiCache ajudar

Seja para fornecer as últimas notícias, um placar dos 10 primeiros colocados, um catálogo de produtos ou a venda de ingressos para um evento, a velocidade é o nome do jogo. O sucesso do seu site e negócios é significativamente afetado pela rapidez com que você fornece o conteúdo.

Em "[Para usuários da Web impacientes, um piscar de olhos é um tempo muito longo para se esperar](#)", os usuários podem registrar uma diferença de 250 milissegundos (1/4 de segundo) entre sites concorrentes. Os usuários tendem a deixar o site mais lento de lado em função do site mais rápido. Os testes feitos na Amazon, citados em [How Webpage Load Time Is Related to Visitor Loss](#) (Como o tempo de carregamento da página está relacionado à perda de visitantes), revelaram que, para cada aumento de 100ms (1/10 segundo) no tempo de carregamento, as vendas diminuiriam 1%.

Se alguém quiser dados, você poderá enviar esses dados com muito mais rapidez se eles estiverem armazenados em cache. Isso é verdade, seja para uma página da Web ou um relatório que desencadeie decisões empresariais. Sua empresa pode se dar ao luxo de não armazenar em cache suas páginas da Web, de modo a entregá-las com a menor latência possível?

Talvez seja intuitivamente óbvio que você queira armazenar em cache seus itens mais solicitados. Mas, por que não armazenar em cache seus itens menos solicitados? Mesmo a consulta de banco de dados ou a chamada de API remota mais otimizada será visivelmente mais lenta do que a recuperação de uma chave simples de um cache de memória. Visivelmente mais lento tende a direcionar os clientes para outros lugares.

Os exemplos a seguir ilustram algumas das maneiras pelas quais o uso ElastiCache pode melhorar o desempenho geral do seu aplicativo.

Tópicos

- [Armazenamento de dados na memória](#)
- [Placares de jogos](#)
- [Mensagens \(publicação/assinatura\)](#)
- [Dados de recomendação \(hashes\)](#)
- [Cache semântico para aplicativos de IA generativa](#)
- [ElastiCache Depoimentos de clientes](#)

Armazenamento de dados na memória

O objetivo principal de um armazenamento de chaves/valores na memória é fornecer acesso ultrarrápido (latência inferior a milissegundos) e de baixo custo a cópias de dados. A maioria dos armazenamentos de dados possui áreas de dados que são frequentemente acessadas, mas raramente atualizadas. Além disso, as consultas a um banco de dados sempre serão mais lentas e mais caras do que localizar uma chave em um cache de pares de chave/valor. Algumas consultas de banco de dados são especialmente caras para executar. Um exemplo é consultas que envolvem junções em várias tabelas ou consultas com cálculos intensos. Ao armazenar em cache esses resultados de consulta, você paga o preço da consulta apenas uma vez. Em seguida, você pode recuperar rapidamente os dados várias vezes sem ter que voltar a executar a consulta.

O que devo armazenar em cache?

Ao decidir quais dados devem ser armazenados em cache, considere estes fatores:

Velocidade e despesas: é sempre mais lento e mais caro obter dados de um banco de dados do que de um cache. Algumas consultas de banco de dados são inerentemente mais lentas e mais caras do que outras. Por exemplo, consultas que executem junções em várias tabelas são muito mais lentas e mais caras do que simples consultas de tabela única. Se os dados de interesse exigirem uma consulta lenta e cara de obter, eles são candidatos para armazenamento em cache. Se a obtenção dos dados exigir uma consulta relativamente rápida e simples, os dados ainda podem ser candidatos para armazenamento em cache, dependendo de outros fatores.

Dados e padrão de acesso: determinar o que armazenar em cache também envolve a compreensão dos dados e seus padrões de acesso. Por exemplo, não faz sentido armazenar dados em cache que mudem rapidamente ou que sejam raramente acessados. Para que o armazenamento em cache ofereça um benefício real, os dados devem ser relativamente estáticos e frequentemente acessados. Um exemplo é um perfil pessoal em um site de mídia social. Por outro lado, você não vai querer armazenar dados em cache se esse armazenamento não oferecer vantagem de velocidade ou custo. Por exemplo, não faz sentido armazenar em cache páginas da Web que retornem resultados de uma pesquisa, pois tais consultas e resultados são quase sempre exclusivos.

Desatualização: por definição, os dados armazenados em cache são dados obsoletos. Mesmo que, em certas circunstâncias, não sejam obsoletos, eles sempre devem ser considerados e tratados como tal. Para dizer se os seus dados são candidatos para armazenamento em cache, determine a tolerância da sua aplicação a dados obsoletos.

Seu aplicativo pode ser capaz de tolerar dados obsoletos em um contexto, mas não em outro. Por exemplo, suponha que o seu site informe um preço de ações negociadas publicamente. Seus

clientes podem aceitar alguma desatualização com um aviso de isenção de responsabilidade de que os preços podem ser n minutos atrasados. Mas, se você informar o preço das ações a um corretor que faça uma venda ou compra, você desejará dados em tempo real.

Considere armazenar seus dados em cache se o seguinte for verdadeiro:

- Seus dados são lentos ou caros para obtenção em comparação à recuperação do cache.
- Os usuários acessam seus dados com frequência.
- Seus dados permanecem relativamente iguais, ou se eles mudam rapidamente, a prontidão não é um problema grande.

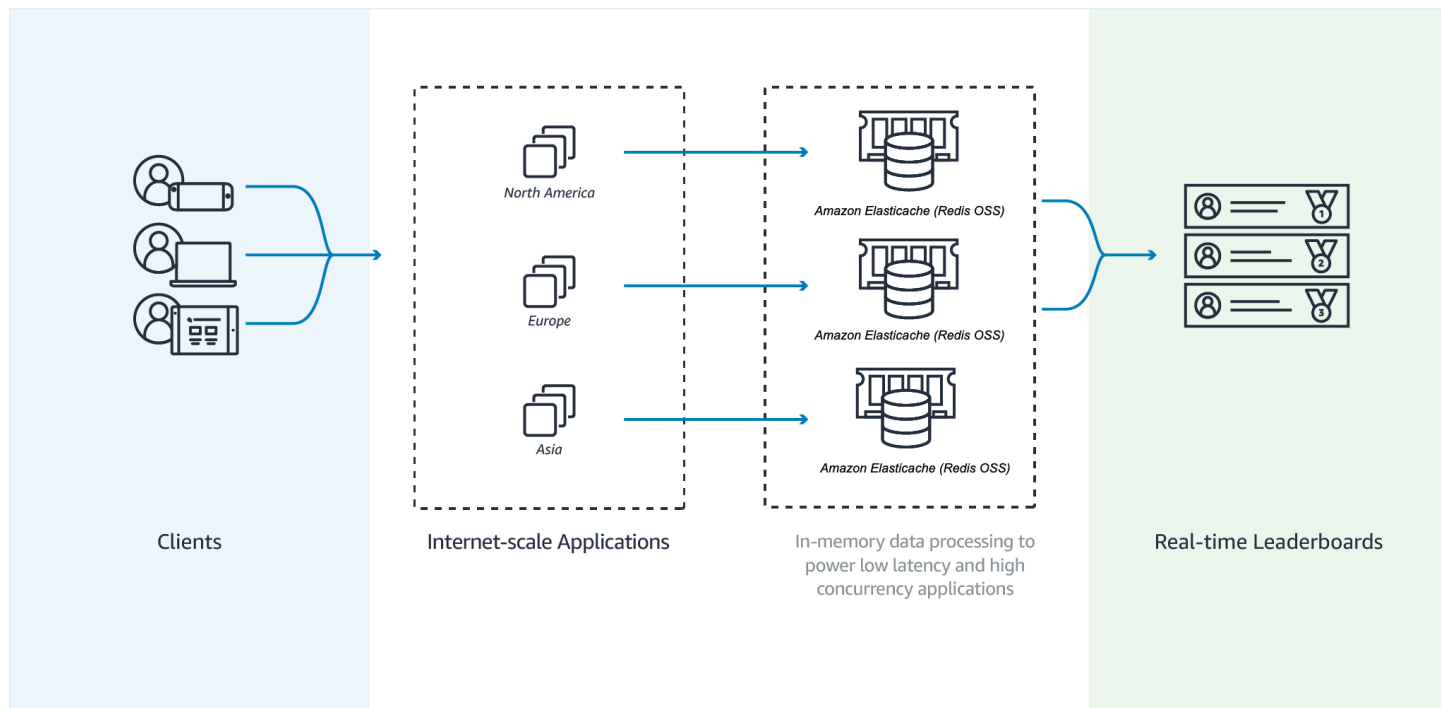
Para obter mais informações, consulte [Estratégias de armazenamento em cache para Memcached](#).

Placares de jogos

Com conjuntos classificados do Valkey ou Redis OSS, você pode mover a complexidade computacional de placares da sua aplicação para o seu cluster.

Placares, como os 10 primeiros líderes em um jogo, são computacionalmente complexos. Isso é especialmente verdadeiro quando há um grande número de jogadores concorrentes e continuamente mudando de pontuação. Os conjuntos classificados do Valkey e do Redis OSS garantem a exclusividade e a ordenação dos elementos. Com os conjuntos classificados, a cada vez que um novo elemento for adicionado ao conjunto classificado ele é reorganizado em tempo real. Em seguida, eles são adicionados ao conjunto em sua ordem numérica correta.

No diagrama a seguir, você pode ver como funciona uma tabela ElastiCache de classificação de jogos.



Example Placa do Valkey ou Redis OSS

Neste exemplo, quatro jogadores e suas pontuações são inseridos em uma lista classificada usando ZADD. O comando ZREVRANGEBYSCORE lista os jogadores por pontuação, de alta a baixa. Em seguida, ZADD é usado para atualizar a pontuação de June, substituindo a entrada existente. Finalmente, ZREVRANGEBYSCORE lista os jogadores por pontuação, de alta a baixa. A lista mostra que June subiu no ranking.

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

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

ZADD leaderboard 232 June

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

- 3) Sandra
- 4) Robert

O seguinte comando informa June sobre sua classificação entre todos os jogadores. Como o ranking é baseado em zero, ZREVRANK retorna um 1 para June, que está na segunda posição.

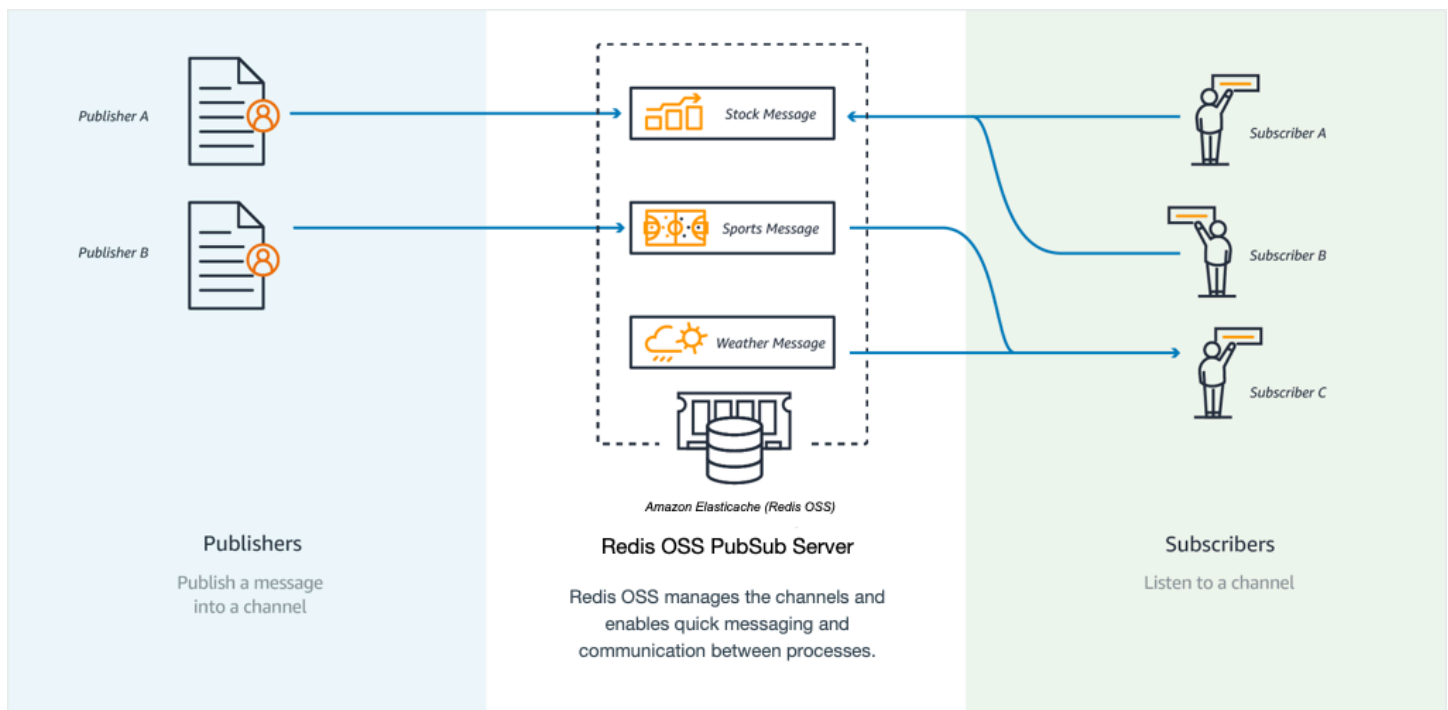
```
ZREVRANK leaderboard June  
1
```

Para obter mais informações, consulte a [documentação do Valkey](#) sobre conjuntos classificados.

Mensagens (publicação/assinatura)

Ao enviar uma mensagem de e-mail, você a envia para um ou mais destinatários especificados. No paradigma de publicação/assinatura do Valkey e Redis OSS, você envia uma mensagem para um canal específico, sem saber quem a receberá. As pessoas que recebem a mensagem são aquelas que assinam o canal. Por exemplo, suponha que você assine o canal news.sports.golf. Você e todos os outros assinantes do canal news.sports.golf receberão todas as mensagens publicadas em news.sports.golf.

A funcionalidade de publicação/assinatura não tem relação com nenhum espaço de chaves. Portanto, ela não interferirá em nenhum nível. No diagrama a seguir, você pode encontrar uma ilustração de ElastiCache mensagens com Valkey e Redis OSS.



Assinatura

Para receber mensagens em um canal, é necessário assiná-lo. Você pode assinar um único canal, vários canais especificados ou todos os canais que corresponderem a um padrão. Para cancelar uma assinatura, você cancela o registro do canal especificado quando se inscreveu nele. Ou, se você se inscreveu usando correspondência de padrões, você cancela a assinatura usando o mesmo padrão usado anteriormente.

Example- Assinatura de um único canal

Para assinar um único canal, use o comando SUBSCRIBE, especificando o canal que deseja assinar. No exemplo a seguir, um cliente assina o canal news.sports.golf.

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

Depois de um tempo, o cliente cancela a assinatura do canal usando o comando UNSUBSCRIBE, especificando o canal do qual cancelar a assinatura.

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

Example- Assinaturas de vários canais especificados

Para assinar vários canais específicos, liste esses canais com o comando SUBSCRIBE. No exemplo a seguir, um cliente assina os canais news.sports.golf, news.sports.soccer e news.sports.skiing.

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

Para cancelar uma assinatura de um canal específico, use o comando UNSUBSCRIBE e especifique o canal do qual cancelar a assinatura.

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

Para cancelar assinaturas em vários canais, use o comando UNSUBSCRIBE e especifique os canais dos quais cancelar a assinatura.

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

Para cancelar todas as assinaturas, use UNSUBSCRIBE e especifique cada canal. Ou use UNSUBSCRIBE e não especifique um canal.

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

or

```
UNSUBSCRIBE
```

Example- Assinaturas usando a correspondência de padrões

Os clientes podem assinar todos os canais que correspondem a um padrão usando o comando PSUBSCRIBE.

No exemplo a seguir, um cliente assina todos os canais de esportes. Você não lista todos os canais de esportes individualmente, como seria feito usando SUBSCRIBE. Em vez disso, com o comando PSUBSCRIBE, você usa a correspondência de padrões.

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

Example Cancelar assinaturas

Para cancelar assinaturas desses canais, use o comando PUNSUBSCRIBE.

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

Important

- A string de canal enviada a um comando [P]SUBSCRIBE e aquela enviada ao comando [P]UNSUBSCRIBE devem corresponder. Você não pode PSUBSCRIBE em news.* e PUNSUBSCRIBE de news.sports.* ou UNSUBSCRIBE de news.sports.golf.
- PSUBSCRIBE e não PUNSUBSCRIBE estão disponíveis para ElastiCache Serverless.

Publicação

Para enviar uma mensagem a todos os assinantes de um canal, use o comando PUBLISH, especificando o canal e a mensagem. O seguinte exemplo publica a mensagem, "It's Saturday and sunny. I'm headed to the links." no canal news.sports.golf.

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


Um cliente não pode publicar em um canal no qual esteja inscrito.

Para obter mais informações, consulte [Publicação/Assinatura](#) na documentação do Valkey.

Dados de recomendação (hashes)

Usar INCR ou DECR no Valkey ou no Redis OSS simplifica a compilação de recomendações. Cada vez que um usuário "curte" um produto, você aumenta um contador de item:productID:like. Cada vez que um usuário "descurte" um produto, você aumenta um contador de item:productID:dislike. Usando hashes, você também pode manter uma lista de todos aqueles que curtiram ou não curtiram um produto.

Example- Curtidas e descurtidas

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

Cache semântico para aplicativos de IA generativa

Operar aplicativos generativos de IA em grande escala pode ser um desafio devido ao custo e à latência associados às chamadas de inferência para grandes modelos de linguagem (). LLMs Você pode usar ElastiCache para armazenamento em cache semântico em aplicativos generativos de IA, permitindo reduzir o custo e a latência das chamadas de inferência do LLM. Com o cache semântico, é possível retornar uma resposta em cache usando a correspondência baseada em vetores para encontrar semelhanças entre os prompts atuais e anteriores. Se o prompt de um usuário for semanticamente parecido com um prompt anterior, uma resposta em cache será retornada em vez de fazer uma nova chamada de inferência do LLM, reduzindo o custo de aplicativos de IA generativa e fornecendo respostas mais rápidas que melhoram a experiência do usuário. Você pode controlar quais consultas são roteadas para o cache configurando limites de similaridade para prompts e aplicando filtros de tag ou metadados numéricos.

As atualizações de índice em tempo real em linha fornecidas pela pesquisa vetorial ElastiCache ajudam a garantir que o cache seja atualizado continuamente à medida que as solicitações do usuário e as respostas do LLM chegam. Essa indexação em tempo real é crucial para manter a atualização dos resultados em cache e das taxas de acerto de cache, especialmente para tráfego intenso. Além disso, ElastiCache simplifica as operações de armazenamento em cache semântico por meio de primitivas de cache maduras, como por chave TTLs, estratégias de despejo configuráveis, operações atômicas e suporte avançado de estrutura de dados e scripts.

Memória para assistentes e atendentes de IA generativa

Você pode usar ElastiCache para fornecer respostas mais personalizadas e sensíveis ao contexto implementando mecanismos de memória que revelam o histórico de conversas entre sessões.

LLMs A memória conversacional permite que assistentes e atendentes de IA generativa retenham e usem interações anteriores para personalizar as respostas e melhorar a relevância. No entanto, simplesmente agregar todas as interações anteriores ao prompt é ineficaz, pois tokens extras irrelevantes aumentam o custo, degradam a qualidade da resposta e correm o risco de exceder a janela de contexto do LLM. Em vez disso, você pode usar a pesquisa vetorial para recuperar e fornecer somente os dados mais relevantes no contexto para cada invocação do LLM.

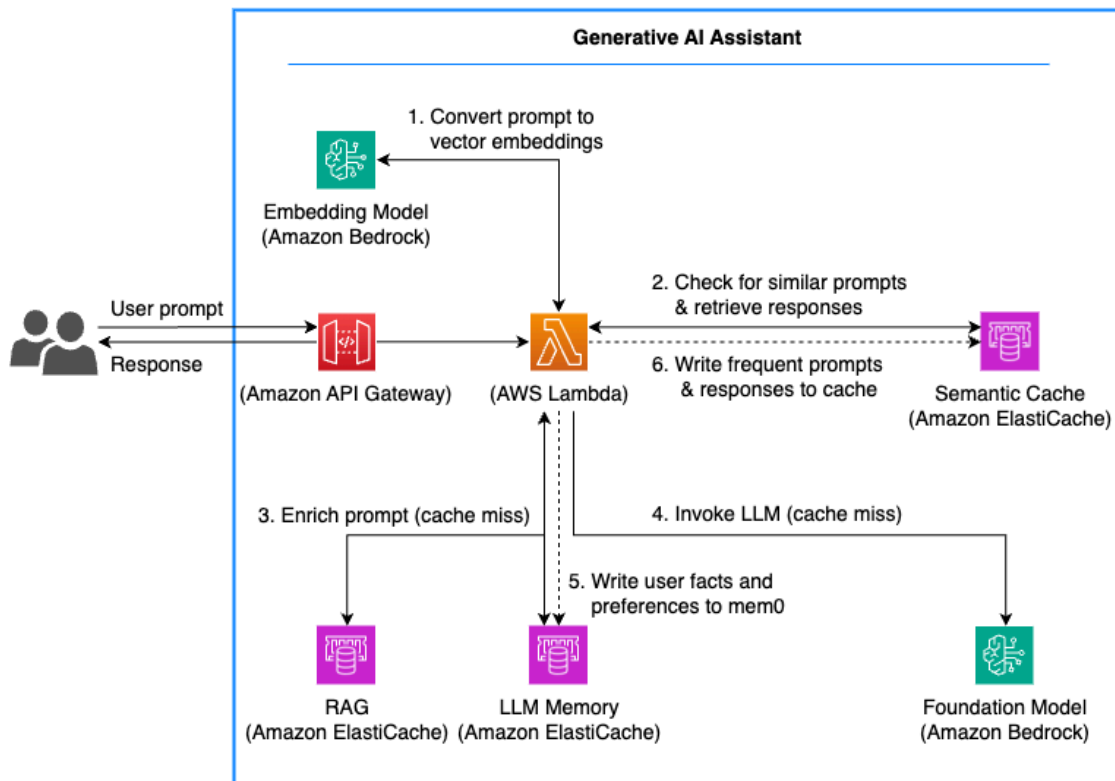
ElastiCache for Valkey fornece integrações com camadas de memória de código aberto, fornecendo conectores integrados para armazenar e recuperar memórias para aplicativos e agentes LLM.

A pesquisa vetorial ElastiCache fornece atualizações rápidas de índices, mantendo a memória atualizada e tornando as novas memórias imediatamente pesquisáveis. A pesquisa vetorial de baixa latência agiliza as pesquisas de memória, permitindo que elas sejam implementadas no caminho online de cada solicitação, não apenas nas tarefas em segundo plano. Além da pesquisa vetorial, ElastiCache o Valkey também fornece primitivos de cache para estado da sessão, preferências do usuário e sinalizadores de recursos, fornecendo um único serviço para armazenar estados de sessão de curta duração e “memórias” de longo prazo em um armazenamento de dados.

Geração aumentada via recuperação (RAG)

O RAG é o processo de LLMs fornecer up-to-date informações prontamente para melhorar a relevância das respostas. O RAG reduz as alucinações e melhora a precisão factual ao aterrar as saídas com fontes de dados do mundo real. Os aplicativos RAG usam a pesquisa vetorial para recuperar conteúdo semanticamente relevante de uma base de conhecimento. A pesquisa vetorial de baixa latência fornecida pelo a ElastiCache torna adequada para implementar RAG em cargas de trabalho que têm grandes conjuntos de dados com milhões de vetores ou mais. Além disso, o suporte para atualizações on-line de índices vetoriais ElastiCache é adequado para assistentes com fluxos de trabalho de upload que precisam garantir que todos os dados enviados sejam imediatamente pesquisáveis. O RAG em sistemas de IA agentes garante que os agentes tenham up-to-date informações para ações precisas. A pesquisa vetorial de baixa latência também é crucial para o RAG em sistemas de IA agêntica, onde uma única consulta pode acionar várias chamadas de LLM e acumular a latência da pesquisa vetorial subjacente.

O diagrama a seguir ilustra um exemplo de arquitetura usada ElastiCache para implementar um cache semântico, mecanismos de memória e RAG para aprimorar um aplicativo de IA generativo em produção.



Pesquisa semântica

A pesquisa vetorial recupera os dados mais relevantes de texto, fala, imagem ou vídeo com base na proximidade do significado ou dos atributos. Esse recurso habilita aplicações de machine learning que dependem da pesquisa por similaridade em diversas modalidades de dados, incluindo mecanismos de recomendação, detecção de anomalias, personalização e sistemas de gerenciamento de conhecimento. Os sistemas de recomendação usam representações vetoriais para capturar padrões complexos no comportamento do usuário e nas características do item, permitindo que eles sugiram o conteúdo mais relevante. A pesquisa vetorial ElastiCache é adequada para esses aplicativos por causa de suas atualizações quase em tempo real e baixa latência, permitindo comparações de similaridade que fornecem recomendações instantâneas e altamente relevantes com base nas interações ao vivo do usuário.

ElastiCache Depoimentos de clientes

Para saber como empresas como Airbnb, PBS, Esri e outras usam a Amazon ElastiCache para expandir seus negócios com uma melhor experiência do cliente, consulte [Como outras pessoas usam](#) a Amazon. ElastiCache

Você também pode assistir aos [vídeos tutoriais](#) para ver outros casos de uso de ElastiCache clientes.

Começando com a Amazon ElastiCache

Use o tutorial prático nesta seção para ajudar você a começar e aprender mais sobre como usar ElastiCache

Tópicos

- [Conf ElastiCacheiguração](#)
- [Criar um cache de tecnologia sem servidor do Valkey](#)
- [Criar um cache de tecnologia sem servidor Redis OSS](#)
- [Criar um cache de tecnologia sem servidor do Memcached](#)
- [Tutoriais: conceitos básicos do Python e do ElastiCache](#)
- [Tutorial: Configurando o Lambda para ElastiCache acesso em uma VPC](#)

Conf ElastiCacheiguração

Para usar o serviço ElastiCache web, siga estas etapas.

Tópicos

- [Inscreva-se para um Conta da AWS](#)
- [Criar um usuário com acesso administrativo](#)
- [Conceder acesso programático](#)
- [Configure suas permissões \(somente para novos ElastiCache usuários\)](#)
- [Configurar EC2](#)
- [Conceder acesso de rede de um grupo de segurança do Amazon VPC para o cache](#)
- [Baixar e configurar o acesso à linha de comando](#)

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.

2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS Centro de Identidade do AWS IAM, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [Console de gerenciamento da AWS](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS.

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o Centro de Identidade do AWS IAM](#) no Guia do usuário do Centro de Identidade do AWS IAM.

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM.

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de logon único ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do Centro de Identidade do AWS IAM.

Conceder acesso programático

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do Console de gerenciamento da AWS. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Recomendado) Use as credenciais do console como	Siga as instruções da interface que deseja utilizar.

Qual usuário precisa de acesso programático?	Para	Por
	credenciais temporárias para assinar solicitações programáticas para o AWS CLI, AWS SDKs ou AWS APIs	• Para o AWS CLI, consulte Login para desenvolvimento AWS local no Guia AWS Command Line Interface do usuário. • Para AWS SDKs isso, consulte Login para desenvolvimento AWS local no Guia de referência de ferramentas AWS SDKs e ferramentas.
Identidade da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI, AWS SDKs, ou AWS APIs	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Configurando o AWS CLI para uso no Centro de Identidade do AWS IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferramentas e AWS APIs, consulte autenticação do IAM Identity Center no Guia de referência de ferramentas AWS SDKs e ferramentas.
IAM	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI, AWS SDKs, ou AWS APIs	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para o AWS CLI, AWS SDKs, ou AWS APIs	Siga as instruções da interface que deseja utilizar. • Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para ferramentas AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de ferramentas AWS SDKs e ferramentas. • Para isso AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Tópicos relacionados:

- [O que é o IAM](#) no Guia do usuário do IAM
- [AWS Credenciais de segurança](#) em referência AWS geral.

Configure suas permissões (somente para novos ElastiCache usuários)

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em Centro de Identidade do AWS IAM:

Crie um conjunto de permissões. Siga as instruções em [Criação de um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM.

- Usuários gerenciados no IAM com provedor de identidades:

Crie um perfil para a federação de identidades. Siga as instruções em [Criando um perfil para um provedor de identidades de terceiros \(federação\)](#) no Guia do Usuário do IAM.

- Usuários do IAM:
 - Crie um perfil que seu usuário possa assumir. Siga as instruções em [Criação de um perfil para um usuário do IAM](#) no Guia do usuário do IAM.
 - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em [Adição de permissões a um usuário \(console\)](#) no Guia do usuário do IAM.

ElastiCache A Amazon cria e usa funções vinculadas a serviços para provisionar recursos e acessar outros AWS recursos e serviços em seu nome. ElastiCache Para criar uma função vinculada ao serviço para você, use a política AWS gerenciada chamada. AmazonElastiCacheFullAccess Essa função é pré-provisionada com uma permissão que o serviço requer para criar uma função vinculada a serviço em seu nome.

Talvez você decida usar uma política gerenciada personalizada, em vez de uma política padrão. Nesse caso, confirme se você tem permissão para chamar `iam:createServiceLinkedRole` ou se criou a função vinculada a serviço do ElastiCache.

Para saber mais, consulte:

- [Criar uma política](#) (IAM)
- [AWS políticas gerenciadas para a Amazon ElastiCache](#)
- [Uso de funções vinculadas ao serviço para o Amazon ElastiCache](#)

Configurar EC2

Você precisará configurar uma EC2 instância a partir da qual se conectará ao seu cache.

- Se você ainda não tem uma EC2 instância, saiba como configurar uma EC2 instância aqui: [Amazon EC2 Getting Started Guide](#).

- Sua EC2 instância deve estar na mesma VPC e ter as mesmas configurações de grupo de segurança do seu cache. Por padrão, a Amazon ElastiCache cria um cache na sua VPC padrão e usa o grupo de segurança padrão. Para seguir este tutorial, certifique-se de que sua EC2 instância esteja na VPC padrão e tenha o grupo de segurança padrão.

Conceder acesso de rede de um grupo de segurança do Amazon VPC para o cache

ElastiCache clusters baseados em nós usam a porta 6379 para comandos Valkey e Redis OSS, e os sem ElastiCache servidor usam a porta 6379 e a porta 6380. Para se conectar e executar com êxito os comandos Valkey ou Redis OSS da sua EC2 instância, seu grupo de segurança deve permitir o acesso a essas portas conforme necessário.

ElastiCache para Memcached usa as portas 11211 e 11212 para aceitar comandos do Memcached. Para se conectar e executar com êxito os comandos do Memcached a partir da sua EC2 instância, seu grupo de segurança deve permitir o acesso a essas portas.

1. Faça login no AWS Command Line Interface e abra o [EC2 console da Amazon](#).
2. No painel de navegação, em Network & Security, escolha Security Groups.
3. Na lista de grupos de segurança, escolha o de segurança para a sua Amazon VPC. A menos que você tenha criado um grupo de segurança para ElastiCache uso, esse grupo de segurança será denominado padrão.
4. Escolha a guia Entrada e, em seguida:

- a. Escolha Editar.
- b. Escolha Adicionar regra.
- c. Na coluna Tipo, escolha Regra TCP personalizada.
- d. Se estiver usando Valkey ou Redis OSS, na caixa Intervalo de portas, digite 6379.

Se estiver usando o Memcached, na caixa Intervalo de portas, digite 11211.

- e. Na caixa Fonte, escolha Qualquer lugar que tenha o intervalo de portas (0.0.0.0/0) para que qualquer EC2 instância da Amazon que você iniciar na sua Amazon VPC possa se conectar ao seu cache.
- f. Se você estiver usando ElastiCache sem servidor, adicione outra regra escolhendo Adicionar regra.
- g. Na coluna Tipo, escolha Regra TCP personalizada.

- h. Se estiver usando ElastiCache para Redis OSS, na caixa Intervalo de portas, digite. 6380

Se estiver usando ElastiCache para Memcached, na caixa Intervalo de portas, digite. 11212

- i. Na caixa Fonte, escolha Qualquer lugar que tenha o intervalo de portas (0.0.0.0/0) para que qualquer EC2 instância da Amazon que você iniciar na sua Amazon VPC possa se conectar ao seu cache.
- j. Escolha Salvar

Baixar e configurar o acesso à linha de comando

Baixe e instale o utilitário valkey-cli.

Se você usa ElastiCache o Valkey, talvez ache útil o utilitário valkey-cli. Se você estiver usando o Redis OSS com redis-cli, considere mudar ElastiCache para valkey-cli, pois ele também funciona para o Redis OSS.

1. Conecte-se à sua EC2 instância da Amazon usando o utilitário de conexão de sua escolha. Para obter instruções sobre como se conectar a uma EC2 instância da Amazon, consulte o [Amazon EC2 Getting Started Guide](#).
2. Baixe e instale o utilitário valkey-cli executando o comando indicado para a configuração.

Amazon Linux 2

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

Note

- Quando você instala o pacote redis6, ele instala o redis6-cli com suporte à criptografia padrão.
- É importante ter suporte de compilação para TLS ao instalar valkey-cli ou redis-cli. ElastiCache O Serverless só pode ser acessado quando o TLS está ativado.

- Se estiver se conectando a um cluster não criptografado, você não precisará da opção `Build_TLS=yes`.

Criar um cache de tecnologia sem servidor do Valkey

Nesta etapa, você cria um novo cache no Amazon ElastiCache.

Console de gerenciamento da AWS

Para criar um novo cache usando o console do ElastiCache:

1. Faça login no Console de gerenciamento da AWS e abra <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Valkey.
3. No lado direito do console, escolha Criar cache do Valkey.
4. Nas Configurações de cache, insira um Nome. Também é possível inserir uma descrição para o cache.
5. Deixe Visualizar configurações padrão selecionado.
6. Clique em Criar para criar o cache.
7. Quando o cache estiver em status “ATIVO”, é possível começar a gravar e ler dados no cache.

AWS CLI

O exemplo de AWS CLI a seguir cria um novo cache usando `create-serverless-cache`.

Linux

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

Windows

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

O valor do campo Status está definido como CREATING.

Para verificar se o ElastiCache terminou a criação do cache, use o comando `describe-serverless-caches`.

Linux

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

Windows

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

Depois de criar o novo cache, avance até [Ler e gravar dados no cache](#).

Ler e gravar dados no cache

Esta seção pressupõe que você criou uma EC2 instância da Amazon e pode se conectar a ela. Para obter instruções sobre como fazer isso, consulte o [Amazon EC2 Getting Started Guide](#).

Esta seção também pressupõe que você tenha definido o acesso à VPC e as configurações do grupo de segurança para EC2 a instância a partir da qual você está se conectando ao cache e configurado valkey-cli na sua instância. EC2 Para obter mais informações sobre essa etapa, consulte [Configurar ElastiCache](#).

Além das etapas abaixo, se você tiver um aplicativo grande ou global, poderá aumentar consideravelmente o desempenho de leitura criando e lendo réplicas. Para obter mais informações sobre essa etapa mais avançada, consulte [Práticas recomendadas para usar réplicas de leitura](#).

Encontre o endpoint do cache

Console de gerenciamento da AWS

Para encontrar o endpoint do seu cache usando o ElastiCache console:

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Valkey.
3. No lado direito do console, clique no nome do cache que você acabou de criar.

4. Nos Detalhes do cache, localize e copie o endpoint do cache.

AWS CLI

O AWS CLI exemplo a seguir mostra como encontrar o endpoint para seu novo cache usando o `describe-serverless-caches` comando. Depois de executar o comando, procure o campo “Endpoint”.

Linux

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

Windows

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

Conectar ao cache do Valkey (Linux)

Agora que você tem o endpoint de que precisa, pode fazer login na sua EC2 instância e se conectar ao cache. No exemplo a seguir, você usa o utilitário `valkey-cli` para se conectar a um cluster. O comando a seguir se conecta a um cache (observação: substitua `cache-endpoint` pelo endpoint que você recuperou na etapa anterior).

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

Conectar ao cache do Valkey (Windows)

Agora que você tem o endpoint de que precisa, pode fazer login na sua EC2 instância e se conectar ao cache. No exemplo a seguir, você usa o utilitário `valkey-cli` para se conectar a um cluster. O comando a seguir conecta a um cache. Abra o prompt de comando, mude para o diretório do Valkey ou Redis OSS e execute o comando (observação: substitua `Cache_Endpoint` pelo endpoint que você recuperou na etapa anterior).

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

```
OK
get a           // Get value for key "a"
"hello"
```

Agora você já pode avançar para [Limpar \(opcional\)](#).

Limpar (opcional)

Se não precisar mais do cache do Amazon ElastiCache criado, você poderá excluí-lo. Esta etapa ajuda a garantir que você não será cobrado pelos recursos que não está utilizando. Você pode usar o console do ElastiCache, a AWS CLI ou a API do ElastiCache para excluir o cache.

Console de gerenciamento da AWS

Para excluir o cache usando o console:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Valkey.
3. Escolha o botão de opção ao lado do cache que você deseja excluir.
4. Selecione Ações no canto superior direito e Excluir.
5. Também é possível optar por fazer um snapshot final antes de excluir o cache.
6. Na tela de confirmação Excluir, reinsira o nome do cache e escolha Excluir para excluir o cluster ou escolha Cancelar para mantê-lo.

Assim que o cache avança para o estado de EXCLUSÃO, você deixa de receber cobranças por ele.

AWS CLI

O exemplo da AWS CLI a seguir exclui um cache usando o comando `delete-serverless-cache`.

Linux

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

Windows

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

```
--serverless-cache-name CacheName
```

O valor do campo Status está definido como EXCLUSÃO.

Agora você já pode avançar para [Próximas etapas](#).

Próximas etapas

Para obter mais informações sobre o ElastiCache, consulte as seguintes páginas:

- [Trabalhando com ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Registros e monitoramento no Amazon ElastiCache](#)
- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)
- [Snapshots e restauração](#)
- [Monitoramento de eventos do ElastiCache pelo Amazon SNS](#)

Criar um cache de tecnologia sem servidor Redis OSS

Nesta etapa, você cria um novo cache no Amazon ElastiCache.

Console de gerenciamento da AWS

Para criar um novo cache usando o console do ElastiCache:

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Redis OSS.
3. No lado direito do console, escolha Criar cache do Redis OSS.
4. Nas Configurações de cache, insira um Nome. Também é possível inserir uma descrição para o cache.
5. Deixe Visualizar configurações padrão selecionado.
6. Clique em Criar para criar o cache.
7. Quando o cache estiver em status “ATIVO”, é possível começar a gravar e ler dados no cache.

AWS CLI

O exemplo de AWS CLI a seguir cria um novo cache usando create-serverless-cache.

Linux

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

Windows

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

O valor do campo Status está definido como CREATING.

Para verificar se o ElastiCache terminou a criação do cache, use o comando describe-serverless-caches.

Linux

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

Windows

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

Depois de criar o novo cache, avance até [Ler e gravar dados no cache](#).

Ler e gravar dados no cache

Esta seção pressupõe que você criou uma EC2 instância da Amazon e pode se conectar a ela. Para obter instruções sobre como fazer isso, consulte o [Amazon EC2 Getting Started Guide](#).

Esta seção também pressupõe que você tenha definido o acesso à VPC e as configurações do grupo de segurança para EC2 a instância a partir da qual você está se conectando ao cache e configurado valkey-cli na sua instância. EC2 Para obter mais informações sobre essa etapa, consulte [Configurar ElastiCache](#).

Encontre o endpoint do cache

Console de gerenciamento da AWS

Para encontrar o endpoint do seu cache usando o ElastiCache console:

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Valkey ou Caches do Redis OSS.
3. No lado direito do console, clique no nome do cache que você acabou de criar.
4. Nos Detalhes do cache, localize e copie o endpoint do cache.

AWS CLI

O AWS CLI exemplo a seguir mostra como encontrar o endpoint para seu novo cache usando o `describe-serverless-caches` comando. Depois de executar o comando, procure o campo "Endpoint".

Linux

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

Windows

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

Conectar ao cache do Valkey ou Redis OSS (Linux)

Agora que você tem o endpoint de que precisa, pode fazer login na sua EC2 instância e se conectar ao cache. No exemplo a seguir, você usa o utilitário `valkey-cli` para se conectar a um cluster. O comando a seguir se conecta a um cache (observação: substitua `cache-endpoint` pelo endpoint que você recuperou na etapa anterior).

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

Conectar ao cache do Valkey ou Redis OSS (Windows)

Agora que você tem o endpoint de que precisa, pode fazer login na sua EC2 instância e se conectar ao cache. No exemplo a seguir, você usa o utilitário valkey-cli para se conectar a um cluster. O comando a seguir conecta a um cache. Abra o prompt de comando, mude para o diretório do Valkey e execute o comando (observação: substitua Cache_Endpoint pelo endpoint que você recuperou na etapa anterior).

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

Agora você já pode avançar para [Limpar \(opcional\)](#).

Limpar (opcional)

Se não precisar mais do cache do Amazon ElastiCache criado, você poderá excluí-lo. Esta etapa ajuda a garantir que você não será cobrado pelos recursos que não está utilizando. Você pode usar o console do ElastiCache, a AWS CLI ou a API do ElastiCache para excluir o cache.

Console de gerenciamento da AWS

Para excluir o cache usando o console:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Valkey ou Redis OSS.
3. Escolha o botão de opção ao lado do cache que você deseja excluir.
4. Selecione Ações no canto superior direito e Excluir.
5. Também é possível optar por fazer um snapshot final antes de excluir o cache.
6. Na tela de confirmação Excluir, reinsira o nome do cache e escolha Excluir para excluir o cluster ou escolha Cancelar para mantê-lo.

Assim que o cache avança para o estado de EXCLUSÃO, você deixa de receber cobranças por ele.

AWS CLI

O exemplo da AWS CLI a seguir exclui um cache usando o comando `delete-serverless-cache`.

Linux

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

Windows

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

O valor do campo `Status` está definido como `EXCLUSÃO`.

Agora você já pode avançar para [Próximas etapas](#).

Próximas etapas

Para obter mais informações sobre o ElastiCache, consulte as seguintes páginas:

- [Trabalhando com ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Registros e monitoramento no Amazon ElastiCache](#)
- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)
- [Snapshots e restauração](#)
- [Monitoramento de eventos do ElastiCache pelo Amazon SNS](#)

Criar um cache de tecnologia sem servidor do Memcached

Console de gerenciamento da AWS

Para criar um novo cache de tecnologia sem servidor do Memcached usando o console do ElastiCache:

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.

2. No painel de navegação, no lado esquerdo do console, escolha Caches do Memcached.
3. No lado direito do console, escolha Criar Clustert Memcached.
4. Nas Configurações de cache, insira um Nome. Também é possível inserir uma descrição para o cache.
5. Deixe Visualizar configurações padrão selecionado.
6. Clique em Criar para criar o cache.
7. Quando o cache estiver em status “ATIVO”, é possível começar a gravar e ler dados no cache.

Para criar um novo cache usando a AWS CLI

O exemplo de AWS CLI a seguir cria um novo cache usando 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
```

O valor do campo Status está definido como CREATING.

Para verificar se o ElastiCache terminou a criação do cache, use o comando describe-serverless-caches.

Linux

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

Windows

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

Depois de criar o novo cache, avance até [Ler e gravar dados no cache](#).

Ler e gravar dados no cache

Esta seção pressupõe que você criou uma EC2 instância da Amazon e pode se conectar a ela. Para obter instruções sobre como fazer isso, consulte o [Amazon EC2 Getting Started Guide](#).

Por padrão, ElastiCache cria um cache na sua VPC padrão. Certifique-se de que sua EC2 instância também seja criada na VPC padrão, para que ela possa se conectar ao cache.

Encontre o endpoint do cache

Console de gerenciamento da AWS

Para encontrar o endpoint do seu cache usando o ElastiCache console:

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Memcached.
3. No lado direito do console, clique no nome do cache que você acabou de criar.
4. Nos Detalhes do cache, localize e copie o endpoint do cache.

AWS CLI

O AWS CLI exemplo a seguir mostra como encontrar o endpoint para seu novo cache usando o `describe-serverless-caches` comando. Depois de executar o comando, procure o campo “Endpoint”.

Linux

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

Windows

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

Conectar usando OpenSSL

Para obter informações sobre como se conectar usando o OpenSSL, consulte [ElastiCache criptografia em trânsito \(TLS\)](#).

Conectar usando o cliente Java do Memcached

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

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

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

Conectar usando o cliente PHP do 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->getLastError_message(), "\n";
}
```

```

    exit(1);
}
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.serverless.us-east-1.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);
?>

```

Conectar usando o cliente Python do Memcached (Pymemcache)

Consulte 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"

```

Conecte-se usando o NodeJS/TS cliente Memcached (Electrode-IO memcache)

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

Instalar por meio de npm `i memcache-client`

Na aplicação, crie um cliente TLS do Memcached assim:

```

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

```

Conectar usando o cliente Rust do Memcached (rust-memcache)

Consulte <https://crates.io/crates/memcache> e <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();
```

Conectar usando o cliente Go do Memcached (Gomemcache)

Veja <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)
```

Conectar usando o cliente Ruby do Memcached (Dalli)

Veja <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")
```

Conecte-se usando o cliente Memcached .NET () EnyimMemcachedCore

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

```
"MemcachedClient": {
  "Servers": [
    {
      "Address": "{cluster_endpoint}",
```

```
"Port": 11211
}
],
"UseSslStream": true
}
```

Agora você já pode avançar para [Limpar \(opcional\)](#).

Limpar (opcional)

Usando o Console de gerenciamento da AWS

O procedimento a seguir exclui um único cache da sua implantação. Para excluir vários caches, repita o procedimento para cada cache que deseja excluir. Você não precisa esperar a finalização da exclusão de um cache antes de iniciar o procedimento para excluir outro.

Para excluir um cache

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache console, escolha o mecanismo em execução no cache que você deseja excluir. É exibida uma lista de todos os caches que executam esse mecanismo.
3. Para escolher o cache a ser excluído, escolha o nome do cache na lista de caches.

Important

Você só pode excluir um cache por vez do ElastiCache console. Escolher vários caches desabilita a operação de exclusão.

4. Em Ações, escolha Excluir.
5. Na tela de confirmação Excluir cache, escolha Excluir para excluir o cache ou Cancelar para mantê-lo.
6. Se você escolheu Excluir, o status do cache muda para excluindo.

Assim que o cache avança para o estado de EXCLUSÃO, você deixa de receber cobranças por ele.

Usando o AWS CLI

O código a seguir exclui o cache my-cache.

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

A ação da delete-serverless-cache CLI exclui somente um cache sem servidor. Para excluir vários caches, chame delete-serverless-cache cada cache sem servidor que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cache sem servidor antes de excluir outro.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Para obter mais informações, consulte o ElastiCache tópico AWS CLI for delete-serverless-cache.

Agora você já pode avançar para [Próximas etapas](#).

Próximas etapas

Para obter mais informações sobre o ElastiCache, consulte:

- [Trabalhando com ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Cotas do ElastiCache](#)
- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)
- [Visualizando ElastiCache eventos](#)

Tutoriais: conceitos básicos do Python e do ElastiCache

Esta seção contém tutoriais práticos para ajudar você a aprender sobre o ElastiCache para Valkey e Redis OSS. Encorajamos você a trabalhar em um dos tutoriais de linguagem específica.

Note

AWSOs SDKs da estão disponíveis para uma ampla variedade de linguagens. Para obter uma lista completa, consulte [Ferramentas para a Amazon Web Services](#).

Tópicos

- [Python e ElastiCache](#)

Python e ElastiCache

Neste tutorial, você usa o AWS SDK para Python (Boto3) para escrever programas simples e realizar as seguintes operações: ElastiCache

- Crie ElastiCache para clusters Redis OSS (modo de cluster ativado e modo de cluster desativado)
- Verifique se existem usuários ou grupos de usuários, caso contrário, crie-os. (Esse atributo está disponível com o Valkey 7.2 e versões posteriores e com o Redis OSS 6.0 a 7.1.)
- Conecte-se a ElastiCache
- Execute operações como definir e obter strings, ler e gravar em streams e publicar e se inscrever no canal. Pub/Sub

Ao trabalhar neste tutorial, você pode consultar a documentação do AWS SDK para Python (Boto). A seção a seguir é específica para ElastiCache: cliente [ElastiCache de baixo nível](#)

Pré-requisitos do tutorial

- Configure uma chave de AWS acesso para usar AWS SDKs o. Para obter mais informações, consulte [Conf ElastiCacheiguração](#).
- Instalar o Python 3.0 ou versões posteriores. Para obter mais informações, consulte <https://www.python.org/downloads>. Para obter instruções, consulte [Início rápido](#) na documentação do Boto 3.

Tópicos

- [Tutorial: Criação de ElastiCache clusters e usuários](#)
- [Tutorial: Conectando-se a ElastiCache](#)
- [Exemplos de uso](#)

Tutorial: Criação de ElastiCache clusters e usuários

Os exemplos a seguir usam o SDK boto3 ElastiCache para operações de gerenciamento do Redis OSS (criação de cluster ou usuário) e redis-py/ para tratamento de dados. redis-py-cluster

Tópicos

- [Criar um cluster com modo cluster desabilitado](#)
- [Criar um cluster com modo cluster desabilitado com TLS e RBAC](#)
- [Criar um cluster com modo cluster habilitado](#)
- [Criar um cluster com modo cluster habilitado com TLS e RBAC](#)
- [Verifique se users/usergroup existe, caso contrário, crie-os](#)

Criar um cluster com modo cluster desabilitado

Copie o programa a seguir e cole-o em um arquivo chamado `CreateClusterModeDisabledCluster.py`.

```
import boto3
import logging

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

def
    create_cluster_mode_disabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumCacheClusters=1,
    cluster', ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled

    Returns a dictionary with the API response

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

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

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

if __name__ == '__main__':

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

    logging.info(elasticacheResponse)
```

Para executar o programa, digite o comando a seguir:

```
python CreateClusterModeDisabledCluster.py
```

Para obter mais informações, consulte [Gerenciamento de clusters no ElastiCache](#).

Criar um cluster com modo cluster desabilitado com TLS e RBAC

Para garantir a segurança, você pode usar Transport Layer Security (TLS) e o Controle de acesso baseado em função (RBAC) ao criar um cluster no modo cluster desabilitado. Ao contrário do AUTH do Valkey ou do Redis OSS, onde todos os clientes autenticados têm acesso total ao grupo de replicação se o token for autenticado, o RBAC permite controlar o acesso ao cluster por meio de grupos de usuários. Esses grupos de usuários são projetados como uma maneira de organizar o acesso a grupos de replicação. Para obter mais informações, consulte [Regras de controle de acesso com base em função \(RBAC\)](#).

Copie o programa a seguir e cole-o em um arquivo chamado `ClusterModeDisabledWithRBAC.py`.

```
import boto3
import logging

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

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

    Returns a dictionary with the API response

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

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

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



```

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

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

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

if __name__ == '__main__':

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

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

    logging.info(response)

```

Para executar o programa, digite o comando a seguir:

```
python ClusterModeDisabledWithRBAC.py
```

Para obter mais informações, consulte [Gerenciamento de clusters no ElastiCache](#).

Criar um cluster com modo cluster habilitado

Copie o programa a seguir e cole-o em um arquivo chamado ClusterModeEnabled.py.

```
import boto3
import logging

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

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

    Returns a dictionary with the API response

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

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

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

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

    return response

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

logging.info(response)
```

Para executar o programa, digite o comando a seguir:

```
python ClusterModeEnabled.py
```

Para obter mais informações, consulte [Gerenciamento de clusters no ElastiCache](#).

Criar um cluster com modo cluster habilitado com TLS e RBAC

Para garantir a segurança, você pode usar Transport Layer Security (TLS) e o Controle de acesso baseado em função (RBAC) ao criar um cluster no modo cluster habilitado. Ao contrário do AUTH do Valkey ou do Redis OSS, onde todos os clientes autenticados têm acesso total ao grupo de replicação se o token for autenticado, o RBAC permite controlar o acesso ao cluster por meio de grupos de usuários. Esses grupos de usuários são projetados como uma maneira de organizar o acesso a grupos de replicação. Para obter mais informações, consulte [Regras de controle de acesso com base em função \(RBAC\)](#).

Copie o programa a seguir e cole-o em um arquivo chamado ClusterModeEnabledWithRBAC.py.

```

import boto3
import logging

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

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

    Returns a dictionary with the API response

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

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

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

```

```

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

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

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

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

    )

    logging.info(response)

```

Para executar o programa, digite o comando a seguir:

```
python ClusterModeEnabledWithRBAC.py
```

Para obter mais informações, consulte [Gerenciamento de clusters no ElastiCache](#).

Verifique se users/usergroup existe, caso contrário, crie-os

Com o RBAC, você cria usuários e atribui a eles permissões específicas usando uma string de acesso. Você atribui os usuários a grupos de usuários alinhados com uma função específica (administradores, recursos humanos) que são então implantados em um ou mais grupos de replicação do Redis ElastiCache OSS. Ao fazer isso, você pode estabelecer limites de segurança entre clientes usando o mesmo grupo ou grupos de replicação do Valkey ou Redis OSS e impedir que os clientes acessem os dados uns dos outros. Para obter mais informações, consulte [Regras de controle de acesso com base em função \(RBAC\)](#).

Copie o programa a seguir e cole-o em um arquivo chamado UserAndUserGroups.py. Atualize o mecanismo para fornecer credenciais. As credenciais neste exemplo são mostradas como substituíveis e atribuídas a um item não declarado. Evite credenciais de codificação rígida.

Este exemplo usa uma string de acesso com as permissões do usuário. Para obter mais informações sobre strings de acesso, consulte [Especificação de permissões usando uma string de acesso](#).

```
import boto3
import logging

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

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

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

```

        else:
            raise

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

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

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

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

    Returns the ARN for the newly created user or the error message
    :param UserId: ElastiCache user ID. User IDs must be unique
    :param UserName: ElastiCache user name. ElastiCache allows multiple users with the
    same name as long as the associated user ID is unique.
    :param Password: Password for user. Must have at least 16 chars.
    :param AccessString: Access string with the permissions for the user.
    :return: user ARN
    """

    try:
        response = client.create_user(
            UserId=UserId,
            UserName=UserName,
            Engine='Redis',
            Passwords=[Password],
            AccessString=AccessString,
            NoPasswordRequired=False
        )

```

```

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

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

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

if __name__ == '__main__':

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

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

```


Para executar o programa, digite o comando a seguir:

```
python UserAndUserGroups.py
```

Tutorial: Conectando-se a ElastiCache

Os exemplos a seguir usam o cliente Valkey ou Redis OSS para se conectar a ElastiCache.

Tópicos

- [Conexão a um cluster com modo cluster desabilitado](#)
- [Conexão a um cluster com modo cluster habilitado](#)

Conexão a um cluster com modo cluster desabilitado

Copie o programa a seguir e cole-o em um arquivo chamado `ConnectClusterModeDisabled.py`. Atualize o mecanismo para fornecer credenciais. As credenciais neste exemplo são mostradas como substituíveis e atribuídas a um item não declarado. Evite credenciais de codificação rígida.

```
from redis import Redis
import logging

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

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

Para executar o programa, digite o comando a seguir:

```
python ConnectClusterModeDisabled.py
```

Conexão a um cluster com modo cluster habilitado

Copie o programa a seguir e cole-o em um arquivo chamado `ConnectClusterModeEnabled.py`.

```
from rediscluster import RedisCluster
import logging
```

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

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

Para executar o programa, digite o comando a seguir:

```
python ConnectClusterModeEnabled.py
```

Exemplos de uso

Os exemplos a seguir usam o SDK boto3 ElastiCache para trabalhar com ElastiCache o Redis OSS.

Tópicos

- [Definir e obter strings](#)
- [Definir e obter um hash com vários itens](#)
- [Publique \(escreva\) e assine \(leia\) em um Pub/Sub canal](#)
- [Gravar e ler de uma stream](#)

Definir e obter strings

Copie o programa a seguir e cole-o em um arquivo chamado SetAndGetStrings.py.

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

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

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

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

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

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

Para executar o programa, digite o comando a seguir:

```
python SetAndGetStrings.py
```

Definir e obter um hash com vários itens

Copie o programa a seguir e cole-o em um arquivo chamado SetAndGetHash.py.

```
import logging
import time

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

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

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

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

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

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

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

Para executar o programa, digite o comando a seguir:

```
python SetAndGetHash.py
```

Publique (escreva) e assine (leia) em um Pub/Sub canal

Copie o programa a seguir e cole-o em um arquivo chamado PubAndSub.py.

```
import logging
import time

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

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

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

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

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

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

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

Para executar o programa, digite o comando a seguir:

```
python PubAndSub.py
```

Gravar e ler de uma stream

Copie o programa a seguir e cole-o em um arquivo chamado ReadWriteStream.py.

```

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

logging.basicConfig(level=logging.INFO)

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

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

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

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

    readerID=threading.currentThread().getName()
    while True:
        try:
            # Check if the stream has any message
            if redis.xlen(streamName)>0:
                # Check if if the messages are new (not acknowledged) or not (already
                processed)
                streamData=redis.xreadgroup(groupName,readerID,
{streamName:'>'},count=1)
                if len(streamData) > 0:
                    msgId,message = streamData[0][1][0]
                    logging.info("{}: Got {} from ID
{}".format(readerID,message,msgId))

```

```

        #Do some processing here. If the message has been processed
        sucessfully, acknowledge it and (optional) delete the message.
        redis.xack(streamName,groupName,msgId)
        logging.info("Stream message ID {} read and processed successfully
        by {}".format(msgId,readerID))
        redis.xdel(streamName,msgId)
    else:
        pass
    except:
        raise

    time.sleep(0.5)

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

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

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

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

```

Para executar o programa, digite o comando a seguir:

```
python ReadWriteStream.py
```

Tutorial: Configurando o Lambda para ElastiCache acesso em uma VPC

Neste tutorial, você pode aprender como criar um cache ElastiCache sem servidor, criar uma função Lambda, testar a função Lambda e, opcionalmente, limpá-la depois.

Tópicos

- [Etapa 1: Criar um cache ElastiCache sem servidor.](#)
- [Etapa 2: criar uma função Lambda para ElastiCache](#)
- [Etapa 3: Teste a função Lambda com ElastiCache](#)
- [Etapa 4: limpar \(opcional\)](#)

Etapa 1: Criar um cache ElastiCache sem servidor.

Para criar um cache de tecnologia sem servidor, siga estas etapas.

Etapa 1.1: criar um cache de tecnologia sem servidor

Nesta etapa, você cria um cache sem servidor na Amazon VPC padrão na região us-east-1 em sua conta usando a (CLI).AWS Command Line Interface Para obter informações sobre como criar cache sem servidor usando o ElastiCache console ou a API, consulte. [Criar um cache de tecnologia sem servidor Redis OSS](#)

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

O valor do campo Status está definido como CREATING. Pode levar um minuto para concluir ElastiCache a criação do cache.

Etapa 1.2: copiar o endpoint do cache de tecnologia sem servidor

Verifique se, ElastiCache para Redis, o OSS terminou de criar o cache com o `describe-serverless-caches` comando.

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

```
--serverless-cache-name cache-01
```

Copie o endereço do endpoint mostrado na saída. Você precisará desse endereço ao criar o pacote de implantação da função do Lambda.

Etapa 1.3: criar o perfil do IAM

1. Crie um documento de política de confiança do IAM, conforme mostrado abaixo, para o perfil que permita que sua conta assuma o novo perfil. Salve a política em um arquivo chamado `trust-policy.json`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::*:role/*"
  },
  {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::*:role/*"
  }]
}
```

2. Crie um documento de política do IAM, conforme mostrado abaixo. Salve a política em um arquivo chamado `policy.json`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],

```



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

3. Criar um perfil do IAM.

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

4. Crie a política do IAM.

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

5. Anexe a política do IAM à função.

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

Etapa 1.4: criar um usuário padrão

1. Crie um novo usuário padrão.

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

2. Crie um novo usuário habilitado para o IAM.

```
aws elasticache create-user \  

```

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

3. Crie um grupo de usuários e anexe o usuário.

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

Etapa 2: criar uma função Lambda para ElastiCache

Para criar uma função Lambda para acessar o ElastiCache cache, siga estas etapas.

Etapa 2.1: criar uma função do Lambda

Neste tutorial, fornecemos um exemplo de código em Python para sua função do Lambda.

Python

O exemplo a seguir, o código Python lê e grava um item no seu ElastiCache cache. Copie o código e o salve em um arquivo chamado `app.py`. Não se esqueça de substituir o valor `elasticache_endpoint` no código pelo endereço do endpoint que você copiou na etapa anterior.

```
from typing import Tuple, Union  
from urllib.parse import ParseResult, urlencode, urlunparse  
  
import boto3.session  
import redis  
from boto3.model import ServiceId  
from boto3.signers import RequestSigner  
from cachetools import TTLCache, cached  
import uuid  
  
class ElastiCacheIAMProvider(redis.CredentialProvider):
```

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

    session = botocore.session.get_session()
    self.request_signer = RequestSigner(
        ServiceId("elasticache"),
        self.region,
        "elasticache",
        "v4",
        session.get_credentials(),
        session.get_component("event_emitter"),
    )

    # Generated IAM tokens are valid for 15 minutes
    @cached(cache=TTLCache(maxsize=128, ttl=900))
    def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
        query_params = {"Action": "connect", "User": self.user}
        if self.is_serverless:
            query_params["ResourceType"] = "ServerlessCache"
        url = urlunparse(
            ParseResult(
                scheme="https",
                netloc=self.cache_name,
                path="/",
                query=urlencode(query_params),
                params="",
                fragment="",
            )
        )
        signed_url = self.request_signer.generate_presigned_url(
            {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
            operation_name="connect",
            expires_in=900,
            region_name=self.region,
        )
        # RequestSigner only seems to work if the URL has a protocol, but
        # ElastiCache only accepts the URL without a protocol
        # So strip it off the signed URL before returning
        return (self.user, signed_url.removeprefix("https://"))

def lambda_handler(event, context):
```

```
username = "iam-user-01" # replace with your user id
cache_name = "cache-01" # replace with your cache name
elasticache_endpoint = "cache-01-xxxxx.serverless.usel.cache.amazonaws.com" #
replace with your cache endpoint
creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
is_serverless=True)
redis_client = redis.Redis(host=elasticache_endpoint, port=6379,
credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

key='uuid'
# create a random UUID - this will be the sample element we add to the cache
uuid_in = uuid.uuid4().hex
redis_client.set(key, uuid_in)
result = redis_client.get(key)
decoded_result = result.decode("utf-8")
# check the retrieved item matches the item added to the cache and print
# the results
if decoded_result == uuid_in:
    print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Valkey.")
else:
    raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

return "Fetched value from Valkey"
```

Esse código usa a biblioteca redis-py do Python para colocar itens no cache e recuperá-los. Esse código usa cachetools para armazenar em cache os tokens IAM Auth gerados por 15 minutos. Para criar um pacote de implantação contendo redis-py e cachetools, realize as etapas a seguir.

No diretório do projeto que contém o arquivo de código-fonte app.py, crie um pacote de pasta no qual instalar as bibliotecas redis-py e cachetools.

```
mkdir package
```

Instale redis-py, cachetools usando pip.

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

Crie um arquivo .zip contendo as bibliotecas redis-py e cachetools. No Linux e no MacOS, execute o comando a seguir. No Windows, use o utilitário zip preferencial para criar um arquivo .zip com as bibliotecas redis-py and cachetools na raiz.

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

Adicione o código de função ao arquivo .zip. No Linux e no MacOS, execute o comando a seguir. No Windows, use o utilitário zip preferencial para adicionar app.py à raiz do arquivo .zip.

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

Etapa 2.2: Criar o perfil do IAM (perfil de execução)

Anexe a política AWS gerenciada nomeada AWSLambdaVPCAccessExecutionRole à função.

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

Etapa 2.3: fazer upload do pacote de implantação (criar função do Lambda)

Nesta etapa, você cria a função Lambda (AccessValkey) usando o comando AWS CLI create-function.

No diretório do projeto que contém o arquivo .zip do pacote de implantação, execute o seguinte comando create-function da CLI do Lambda.

Para a opção de perfil, use o ARN da função de execução criada na etapa anterior. Para vpc-config, insira listas separadas por vírgulas das sub-redes da VPC padrão e o ID do grupo de segurança da VPC padrão. É possível encontrar esses valores no console do Amazon VPC. Para encontrar as sub-redes da sua VPC padrão, escolha Sua e, em seguida VPCs, escolha a VPC padrão AWS da sua conta. Para localizar o grupo de segurança dessa VPC, vá para Segurança e escolha Grupos de segurança. Não se esqueça de selecionar a região us-east-1.

```
aws lambda create-function \
  --function-name AccessValkey \
  --region us-east-1 \
  --zip-file fileb://my_deployment_package.zip \
  --role arn:aws:iam::123456789012:role/elasticache-iam-auth-app \
  --handler app.lambda_handler \
  --runtime python3.12 \
  --timeout 30 \
```

```
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-  
security-group-id
```

Etapa 3: Teste a função Lambda com ElastiCache

Nesta etapa, você invoca a função do Lambda manualmente usando o comando de invocação. Quando a função Lambda é executada, ela gera um UUID e o grava no ElastiCache cache que você especificou no seu código Lambda. Depois, a função do Lambda recupera o item do cache.

1. Invoque a função Lambda AccessValkey () usando AWS Lambda o comando invoke.

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

2. Verifique se a função do Lambda foi executada com êxito, da seguinte forma:

- Analise o arquivo output.txt.
- Verifique os resultados em CloudWatch Logs abrindo o CloudWatch console e escolhendo o grupo de registros para sua função (/aws/lambda/AccessValkey). O fluxo de logs deve conter uma saída semelhante à mostrada a seguir:

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

- Analise os resultados no AWS Lambda console.

Etapa 4: limpar (opcional)

Para limpar, siga estas etapas.

Etapa 4.1: excluir a função do Lambda

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

Etapa 4.2: excluir o cache de tecnologia sem servidor

Excluir o cache.

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

Remover usuários e grupos de usuários.

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

Etapa 4.3: remover políticas e o perfil do IAM

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

Criação e gerenciamento de um cluster baseado em nós ElastiCache

Se você precisar de um controle refinado sobre seu ElastiCache cluster, poderá criar um cluster baseado em nós. ElastiCache permite que você opere um cluster baseado em nós escolhendo o tipo de nó, o número de nós e o posicionamento dos nós nas zonas de AWS disponibilidade do seu cluster. Como ElastiCache é um serviço totalmente gerenciado, ele gerencia automaticamente o provisionamento de hardware, o monitoramento, as substituições de nós e a aplicação de patches de software para seu cluster.

Para obter informações sobre como configurar, consulte [Conf ElastiCacheiguração](#). Para obter detalhes sobre como gerenciar, atualizar ou excluir nós ou clusters, consulte [Gerenciamento de nós no ElastiCache](#). Para uma visão geral dos principais componentes de uma ElastiCache implantação da Amazon ao criar um ElastiCache cluster baseado em nós, consulte esses [conceitos-chave](#).

Tópicos

- [ElastiCache componentes e recursos](#)
- [ElastiCache terminologia](#)
- [Tutorial: Como criar um cluster baseado em nós ElastiCache](#)
- [Excluir um cluster](#)
- [Outros tutoriais e vídeos do ElastiCache](#)
- [Gerenciamento de nós no ElastiCache](#)
- [Gerenciamento de clusters no ElastiCache](#)
- [Comparação de clusters Valkey, Memcached e Redis OSS baseados em nós](#)
- [Migração on-line para o Valkey ou Redis OSS](#)
- [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#)

ElastiCache componentes e recursos

A seguir, você encontrará uma visão geral dos principais componentes de uma ElastiCache implantação da Amazon.

Tópicos

- [ElastiCache nós](#)
- [ElastiCache cacos](#)
- [ElastiCache aglomerados](#)
- [ElastiCache replicação](#)
- [ElastiCache endpoints](#)
- [ElastiCache grupos de parâmetros](#)
- [ElastiCache segurança](#)
- [ElastiCache grupos de sub-redes](#)
- [ElastiCache backups](#)
- [ElastiCache eventos](#)

ElastiCache nós

Um nó é o menor componente de uma ElastiCache implantação. Um nó pode existir isoladamente ou em algum relacionamento com outros nós.

Um nó é um bloco de tamanho fixo da RAM em armazenamento anexado a rede seguro. Cada nó executa uma instância do mecanismo e da versão escolhidos ao criar o cluster. Se necessário, você pode expandir ou reduzir os nós em um cluster para um tipo de instância diferente. Para obter mais informações, consulte [Dimensionamento ElastiCache](#).

Cada nó dentro de um cluster é do mesmo tipo de instância e executa o mesmo mecanismo de cache. Cada nó de cache possui seu próprio nome DNS (Serviço de Nomes de Domínio) e porta. Vários tipos de nós de cache são suportados, cada um com quantidades diversificadas de memória associada. Para obter uma lista dos tipos de instâncias de nó, consulte [Tipos de nó compatíveis](#).

Você pode comprar nós em uma pay-as-you-go base, pagando apenas pelo uso de um nó. Ou você pode comprar nós reservados a uma taxa por hora bastante reduzida. Se a sua taxa de uso for alta, a compra de nós reservados poderá economizar dinheiro. Suponha que seu cluster está quase sempre em uso e você ocasionalmente adiciona nós para lidar com picos de uso. Nesse caso, você pode comprar uma série de nós reservados para execução na maior parte do tempo. Em seguida, você pode comprar pay-as-you-go nós nos momentos em que precisar adicionar nós ocasionalmente. Para obter mais informações sobre nós reservados, consulte [Nós reservados](#).

Para obter mais informações sobre nós, consulte [Gerenciamento de nós no ElastiCache](#).

ElastiCache cacos

Um fragmento do Valkey ou Redis OSS (chamado de grupo de nós na API e na CLI) é um agrupamento de um a seis nós relacionados. Um cluster do Valkey ou Redis OSS com modo de cluster habilitado sempre tem pelo menos um fragmento.

A fragmentação é um método de particionamento de banco de dados que separa grandes bancos de dados em partes menores, mais rápidas e mais fáceis de gerenciar, chamadas fragmentos de dados. Isso pode aumentar a eficiência do banco de dados distribuindo as operações em várias seções separadas. O uso de fragmentos pode oferecer muitos benefícios, incluindo melhor desempenho, escalabilidade e eficiência de custos.

Os clusters Valkey e Redis OSS com o modo de cluster habilitado podem ter até 500 fragmentos, com seus dados particionados entre os fragmentos. O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo Valkey ou Redis OSS for 5.0.6 ou posterior. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#). Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Um fragmento de vários nós implementa a replicação por ter um nó primário de leitura/gravação e de 1 a 5 nós de réplicas. Para obter mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

Para obter mais informações sobre fragmentos, consulte [Operação com fragmentos no ElastiCache](#).

ElastiCache aglomerados

Um cluster é um agrupamento lógico de um ou mais [nós](#). Os dados são particionados entre os nós em um cluster Memcached e entre os fragmentos em um cluster Valkey ou Redis OSS que tem o modo de cluster habilitado.

Muitas ElastiCache operações são direcionadas a clusters:

- Criação de um cluster
- Modificar um cluster
- Tirar snapshots de um cluster (todas as versões do Redis)
- Excluir um cluster
- Visualizar os elementos em um cluster
- Adicionar ou remover tags de alocação de custos para e de um cluster

Para obter informações mais detalhadas, consulte os seguintes tópicos relacionados:

- [Gerenciamento de clusters no ElastiCache](#) e [Gerenciamento de nós no ElastiCache](#)

Informações sobre clusters, nós e operações relacionadas.

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

Informações sobre ElastiCache limites, como o número máximo de nós ou clusters. Para exceder alguns desses limites, você pode fazer uma solicitação usando o [formulário de solicitação de nós de ElastiCache cache da Amazon](#).

- [Atenuar falhas](#)

Informações sobre como melhorar a tolerância a falhas dos seus clusters e grupos de replicação Valkey ou Redis OSS.

Configurações de cluster típicas

Veja a seguir configurações de cluster típicas.

Clusters Valkey ou Redis OSS

Os clusters Valkey ou Redis OSS com o modo de cluster desabilitado sempre contêm apenas um fragmento (na API e na CLI, um grupo de nós). Um fragmento Valkey ou Redis OSS contém de um a seis nós. Se houver mais de um nó em um fragmento, este oferecerá suporte para a replicação. Nesse caso, um nó é o nó read/write primário e os outros são nós de réplica somente para leitura.

Para melhorar a tolerância a falhas, recomendamos ter pelo menos dois nós em um cluster do Valkey ou Redis OSS e habilitar o Multi-AZ. Para obter mais informações, consulte [Atenuar falhas](#).

Conforme a demanda do seu cluster Valkey ou Redis OSS muda, você pode diminuir ou aumentar a escala verticalmente. Para fazer isso, mova seu cluster para um tipo de instância de nó diferente.

Se sua aplicação exigir muita leitura, recomendamos adicionar réplicas somente leitura ao cluster. Ao fazer isso, você pode distribuir as leituras entre um número mais apropriado de nós.

Você também pode usar a classificação de dados em níveis. Os dados acessados mais frequentemente são armazenados em memória e os dados acessados menos frequentemente são armazenados em disco. A vantagem de usar a classificação de dados em níveis é que ela diminui os requisitos de memória. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

ElastiCache suporta a alteração dinâmica do tipo de nó de um cluster Valkey ou Redis OSS para um tipo de nó maior. Para obter informações sobre como expandir ou reduzir a escala, consulte [Escalonamento para Valkey ou Redis OSS \(modo cluster desabilitado\)](#) ou [Escalar nós de réplica para Valkey ou Redis OSS \(modo cluster desativado\)](#).

Configurações de cluster típicas para Memcached

O Memcached suporta até 300 nós por cliente em cada AWS região, com cada cluster tendo de 1 a 60 nós. Você pode particionar seus dados entre os nós em um cluster Memcached.

Quando você executa o mecanismo do Memcached, os clusters podem ser formados por de 1 a 60 nós. Você particiona seu banco de dados entre os nós. Seu aplicativo lê e grava no endpoint de cada nó. Para obter mais informações consulte [Descoberta automática](#).

Para melhorar a tolerância a falhas, localize seus nós do Memcached em várias zonas de disponibilidade (AZs) dentro da região do AWS cluster. Dessa forma, uma falha em uma AZ terá um impacto mínimo em todo o seu cluster e no aplicativo. Para obter mais informações, consulte [Atenuar falhas](#).

À medida que a demanda em seu cluster Memcached mudar, você poderá expandir ou reduzir adicionando ou removendo nós, e isso reparticiona seus dados entre o novo número de nós. Ao particionar seus dados, recomendamos usar o hashing consistente. Para obter mais informações sobre o hashing consistente, consulte [Configurando seu ElastiCache cliente para balanceamento de carga eficiente \(Memcached\)](#).

ElastiCache replicação

Para o Valkey e Redis OSS, a replicação é implementada agrupando de dois a seis nós em um fragmento (na API e CLI, chamado de grupo de nós). Um desses nós é o nó primário de leitura/gravação. Todos os outros nós são nós de réplica somente leitura. As replicações estão disponíveis somente ElastiCache para Valkey e Redis OSS, e não para Memcached. ElastiCache

Cada nó réplica mantém uma cópia dos dados do nó primário do cluster. Os nós de réplica usam mecanismos de replicação assíncronos para se manterem sincronizados com o nó primário. Os aplicativos podem ler de qualquer nó no cluster, mas podem gravar apenas em nós primários. As réplicas de leitura aumentam a escalabilidade, distribuindo as leituras entre vários endpoints. As réplicas de leitura também melhoram a tolerância a falhas, mantendo várias cópias dos dados. Localizar réplicas de leitura em várias zonas de disponibilidade melhora ainda mais a tolerância a falhas. Para obter mais informações sobre a tolerância a falhas, consulte [Atenuar falhas](#).

Os clusters Valkey ou Redis OSS oferecem suporte para um único fragmento (na API e CLI, chamado de grupo de nós).

A replicação sob a perspectiva da API e da CLI usa uma terminologia diferente para manter a compatibilidade com as versões anteriores, mas os resultados são os mesmos. A tabela a seguir mostra os termos da API e da CLI para implementar a replicação.

Comparando replicação: Valkey ou Redis OSS (modo cluster desabilitado) e Valkey ou Redis OSS (modo cluster habilitado)--> Cluster Valkey ou Redis OSS com modo cluster habilitado vs. Cluster Valkey ou Redis OSS com modo cluster desabilitado

Na tabela a seguir, você pode encontrar uma comparação dos recursos dos grupos de replicação Valkey ou Redis OSS (modo cluster desabilitado) e Valkey ou Redis OSS (modo cluster habilitado).

	Cluster Valkey ou Redis OSS com modo de cluster desabilitado	Cluster Valkey ou Redis OSS com modo de cluster habilitado
Fragmentos (grupos de nós)	1	1–500
Réplicas para cada fragmento (grupo de nós)	0–5	0–5
Particionamento de dados	Não	Sim
Adiciona/Exclui réplicas	Sim	Sim
Adiciona/Exclui grupos de nós	Não	Sim
Suporte a aumento de escala na horizontal	Sim	Sim

	Cluster Valkey ou Redis OSS com modo de cluster desabilitado	Cluster Valkey ou Redis OSS com modo de cluster habilitado
Suporte a atualizações do mecanismo	Sim	Sim
Promove réplica para primário	Sim	Automatico
multi-AZ	Opcional	Obrigatório
Restaura backup	Sim	Sim

Observações:

Se algum primário não tiver réplicas e o primário falhar, você perderá todos os dados desse primário.

Você pode usar o backup e a restauração para migrar para Valkey ou Redis OSS (modo cluster habilitado).

Você pode usar o backup e a restauração para redimensionar seu cluster do Valkey ou Redis OSS (modo cluster habilitado).

Todos os fragmentos (na API e CLI, grupos de nós) e grupos de nós devem residir na mesma região da AWS. No entanto, você pode provisionar os nós individuais em várias zonas de disponibilidade dentro dessa AWS região.

Réplicas de leitura protegem contra a perda de dados em potencial, pois seus dados são replicados em dois ou mais nós: o primário e uma ou mais réplicas de leitura. Para maior confiabilidade e recuperação mais rápida, recomendamos que você crie uma ou mais réplicas de leitura em diferentes zonas de disponibilidade.

Você também pode aproveitar os datastores globais. Ao usar o recurso Global Datastore for Redis OSS, você pode trabalhar com replicação totalmente gerenciada, rápida, confiável e segura em todas as regiões.AWS Usando esse recurso, você pode criar clusters de réplicas de leitura entre regiões ElastiCache para permitir leituras de baixa latência e recuperação de desastres em todas as regiões.AWS Para obter mais informações, consulte [Replicação entre AWS regiões usando armazenamentos de dados globais](#).

Replicação: limites e exclusões

- O Multi-AZ não tem suporte em tipos de nó T1.

ElastiCache endpoints

Um endpoint é o endereço exclusivo que seu aplicativo usa para se conectar a um ElastiCache nó ou cluster.

Endpoints de nó único para Valkey ou Redis OSS com modo de cluster desabilitado

O endpoint para um cluster Valkey ou Redis OSS de nó único é usado para se conectar ao cluster para leituras e gravações.

Endpoints de vários nós para Valkey ou Redis OSS com modo de cluster desabilitado

Um cluster Valkey ou Redis OSS de vários nós com modo de cluster desabilitado tem dois tipos de endpoints. O endpoint primário sempre se conecta ao nó primário no cluster, mesmo que o nó específico na função primária mude. Use o endpoint primário para todas as gravações no cluster.

Use o Endpoint de leitor para dividir uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura. Use os pontos finais individuais do Node para operações de leitura (neles API/CLI são chamados de pontos finais de leitura).

Endpoints Valkey ou Redis OSS (modo cluster habilitado)

Um cluster Valkey ou Redis OSS com modo de cluster habilitado tem um único endpoint de configuração. Ao se conectar ao endpoint de configuração, a aplicação pode descobrir os endpoints primários e os endpoints de leitura para cada fragmento no cluster.

Para obter mais informações, consulte [Encontrar endpoints de conexão no ElastiCache](#).

ElastiCache para endpoints Memcached

Cada nó em um cluster Memcached tem seu próprio endpoint. O cluster também possui um endpoint chamado de endpoint de configuração. Se você habilitar a Descoberta automática e se conectar ao endpoint de configuração, seu aplicativo conhecerá automaticamente cada endpoint de nó, mesmo depois de adicionar ou remover nós do cluster. Para obter mais informações consulte [Descoberta automática](#).

Para obter mais informações, consulte [Encontrar endpoints de conexão no ElastiCache](#).

ElastiCache grupos de parâmetros

Parameter groups de cache são uma maneira fácil de gerenciar as configurações de tempo de execução do software do mecanismo com suporte. Os parâmetros são usados para controlar o uso de memória, as políticas de remoção, o tamanho dos itens e muito mais. Um grupo de ElastiCache parâmetros é uma coleção nomeada de parâmetros específicos do mecanismo que você pode aplicar a um cluster. Ao fazê-lo, você garante que todos os nós do cluster sejam configurados exatamente da mesma forma.

Para obter uma lista de parâmetros com suporte, seus valores padrão e quais podem ser modificados, consulte [DescribeEngineDefaultParameters](#) (ILC: [describe-engine-default-parameters](#)).

Para obter informações mais detalhadas sobre grupos de ElastiCache parâmetros, consulte [Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache](#).

ElastiCache segurança

Para maior segurança, o acesso aos ElastiCache nós é restrito aos aplicativos em execução nas EC2 instâncias da Amazon que você permite. Você pode controlar as EC2 instâncias da Amazon que podem acessar seu cluster usando grupos de segurança.

Por padrão, todos os novos ElastiCache clusters são lançados em um ambiente Amazon Virtual Private Cloud (Amazon VPC). Você pode usar grupos de sub-redes para conceder acesso ao cluster a partir de EC2 instâncias da Amazon em execução em sub-redes específicas.

Além de restringir o acesso ao nó, ElastiCache oferece suporte a TLS e criptografia local para nós que executam versões especificadas do. ElastiCache Para saber mais, consulte:

- [Segurança de dados no Amazon ElastiCache](#)
- [Autenticar com o comando AUTH do Valkey e Redis OSS](#)

ElastiCache grupos de sub-redes

Um grupo de sub-redes é uma coleção de sub-redes (normalmente privadas) que você pode designar para clusters em execução em um ambiente de Amazon VPC.

Se você criar um cluster em uma VPC da Amazon, deverá especificar um grupo de sub-rede de cache. O ElastiCache usa esse grupo de sub-redes de cache para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós de cache.

Para obter mais informações sobre o uso de grupos de sub-redes de cache em um ambiente da Amazon VPC, consulte o seguinte:

- [Amazon VPCs e ElastiCache segurança](#)
- [Etapa 3. Autorizar o acesso ao cluster](#)
- [Sub-redes e grupos de sub-redes](#)

ElastiCache backups

Um backup é uma point-in-time cópia de um cluster OSS Valkey ou Redis ou cache sem servidor, ou um cache sem servidor Memcached. Backups podem ser usados para restaurar um cluster existente ou para propagar um novo cluster. Backups consistem em todos os dados de um cluster mais alguns metadados.

Dependendo da versão do Valkey ou Redis OSS em execução no seu cluster, o processo de backup exige diferentes quantidades de memória reservada para ter sucesso. Para saber mais, consulte:

- [Snapshots e restauração](#)
- [Como a sincronização e o backup são implementados](#)
- [Impacto sobre o desempenho dos backups de clusters baseados em nós](#)
- [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#)

ElastiCache eventos

Quando eventos importantes acontecem em um cluster, ElastiCache envia uma notificação para um tópico específico do Amazon SNS. Esses eventos podem incluir fatores como falha ou êxito ao adicionar um nó, modificação de um grupo de segurança, etc. Ao monitorar eventos importantes, você pode se manter informado sobre o estado atual de seus clusters e, em vários casos, executar a ação corretiva.

Para obter mais informações sobre ElastiCache eventos, consulte [Monitoramento de eventos do ElastiCache pelo Amazon SNS](#).

ElastiCache terminologia

Em outubro de 2016, a Amazon ElastiCache lançou o suporte para o Redis OSS 3.2. Nesse ponto, adicionamos suporte para particionar seus dados em até 500 fragmentos (chamados de grupos de nós na ElastiCache API e).AWS CLI Para preservar a compatibilidade com as versões anteriores, ampliamos as operações da API versão 2015-02-02 para incluir a nova funcionalidade do Redis OSS.

Ao mesmo tempo, começamos a usar a terminologia no ElastiCache console que é usada nessa nova funcionalidade e é comum em todo o setor. Essas mudanças significam que, em alguns pontos, a terminologia usada na API e na CLI pode ser diferente da terminologia usada no console. A lista a seguir identifica termos que podem diferir entre a API e a CLI e o console.

Cluster ou nó de cache vs. nó

Há uma one-to-one relação entre um nó e um cluster quando não há nós de réplica. Assim, o ElastiCache console costumava usar os termos de forma intercambiável. O console agora usa apenas o termo nó. A única exceção é o botão Create Cluster, que executa o processo para criar um cluster com ou sem nós de réplica.

A ElastiCache API e AWS CLI continue usando os termos como usavam no passado.

Cluster versus grupo de replicação do Valkey ou Redis OSS

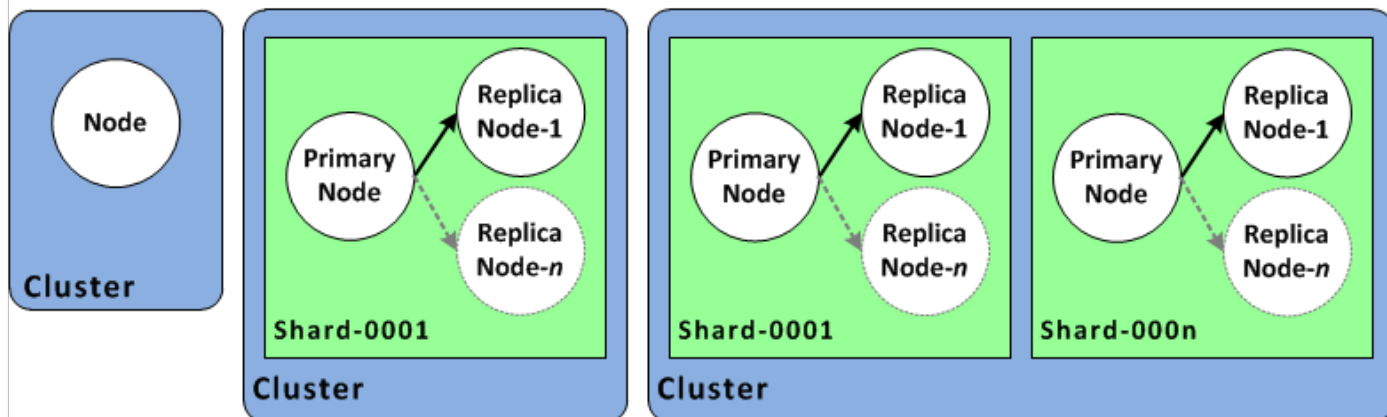
O console agora usa o termo cluster ElastiCache para todos os clusters Redis OSS. O console usa o termo Cluster em todas essas circunstâncias:

- Quando o cluster é um cluster Valkey ou Redis OSS de nó único.
- Quando o cluster é um do Valkey ou Redis OSS (modo cluster desabilitado) com suporte para replicação dentro de um único fragmento (na API e CLI, chamado de grupo de nós).
- Quando o cluster é um do Valkey ou Redis (modo cluster habilitado) que oferece suporte à replicação em 1-90 fragmentos ou até 500 com uma solicitação de aumento de limite. Para solicitar um aumento de limite, consulte [Limites do serviço da AWS](#) e escolha o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Para obter mais informações sobre grupos de replicação do Valkey ou do Redis OSS, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

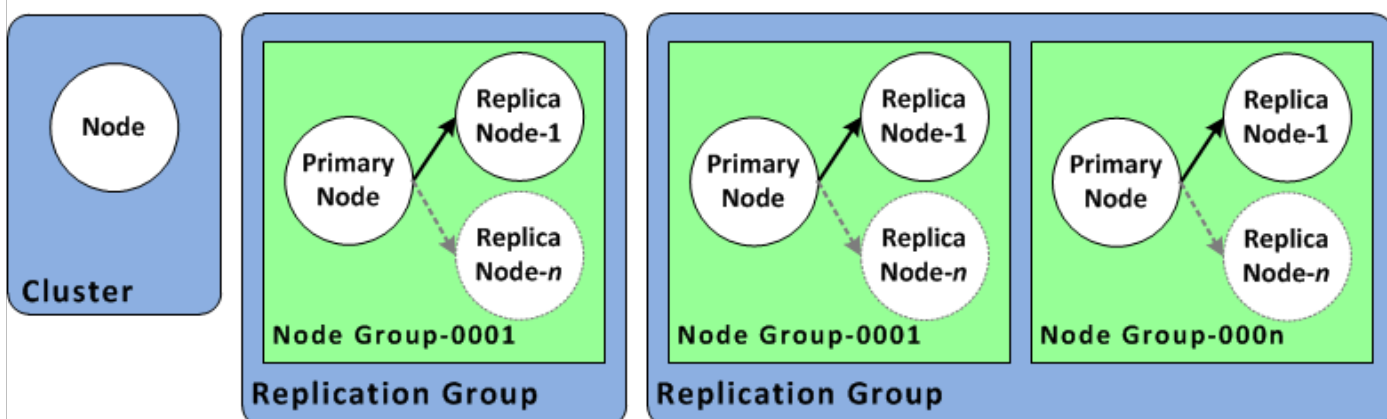
O diagrama a seguir ilustra as várias topologias dos ElastiCache clusters OSS do Redis a partir da perspectiva do console.

ElastiCache (Redis OSS): Console View



A ElastiCache API e AWS CLI as operações ainda distinguem um único nó ElastiCache para clusters Redis OSS de grupos de replicação Valkey ou Redis OSS de vários nós. O diagrama a seguir ilustra as várias ElastiCache topologias do Redis OSS a partir da API e da ElastiCache perspectiva.AWS CLI

ElastiCache (Redis OSS): API/CLI View



Grupo de replicação do Valkey ou do Redis OSS versus datastore global

Um datastore global é uma coleção de um ou mais clusters que são replicados entre regiões, enquanto um grupo de replicação do Valkey ou Redis OSS replica dados em um cluster habilitado para modo cluster com vários fragmentos. Um datastore global consiste no seguinte:

- **Cluster primário (ativo):** um cluster primário aceita gravações que são replicadas para todos os clusters dentro do datastore global. Um cluster primário também aceita solicitações de leitura.
- **Cluster secundário (passivo):** um cluster secundário só aceita solicitações de leitura e replica as atualizações de dados de um cluster primário. Um cluster secundário precisa estar em uma AWS região diferente do cluster primário.

Para obter informações sobre datastores globais, consulte [Replicação entre AWS regiões usando datastores globais](#).

Tutorial: Como criar um cluster baseado em nós ElastiCache

Veja como criar um ElastiCache cluster baseado em nós para Valkey, Memcached e Redis OSS.

Tópicos

- [Criação de um ElastiCache cluster baseado em nós para o Valkey](#)
- [Criação de um ElastiCache cluster baseado em nós para o Redis OSS](#)

Criação de um ElastiCache cluster baseado em nós para o Valkey

A seguir estão as ações únicas que você deve realizar para criar um ElastiCache cluster baseado em nós para o Valkey.

Etapa 1: criar um grupo de sub-redes

Antes de criar um cluster ElastiCache (Valkey), primeiro você cria um grupo de sub-redes. Um grupo de sub-redes de cache é uma coleção de sub-redes que você pode querer designar aos seus clusters em uma VPC. Ao executar um cluster em uma VPC, é necessário selecionar um grupo de sub-redes de cache. Em seguida, ElastiCache usa esse grupo de sub-redes de cache para atribuir endereços IP dentro dessa sub-rede a cada nó de cache no cluster.

Quando você criar um novo grupo de sub-redes, observe o número de endereços IP disponíveis. Se a sub-rede tiver muito poucos endereços IP livres, talvez haja um limite no que diz respeito ao número de nós adicionais que é possível acrescentar ao cluster. Para resolver esse problema, você pode atribuir uma ou mais sub-redes a um grupo de sub-redes para ter um número suficiente de endereços IP na zona de disponibilidade do seu cluster. Depois disso, você pode adicionar mais nós ao seu cluster.

Para obter mais informações sobre a configuração, ElastiCache consulte [Conf ElastiCacheiguração](#).

Os procedimentos a seguir mostram como criar um grupo de sub-rede chamado mysubnetgroup (console) e a AWS CLI.

Criação de um grupo de sub-redes (console)

O procedimento a seguir mostra como criar um grupo de sub-redes (console).

Como criar um grupo de sub-redes (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha Subnet Groups (Grupos de sub-redes).
3. Selecione Create Subnet Group (Criar grupo de sub-redes).
4. No assistente Create Subnet Group (Criar grupo de sub-redes), faça o seguinte. Quando estiver satisfeito com todas as configurações, clique em Yes, Create (Sim, criar).
 - a. Na caixa Name (Nome), digite um nome para o seu grupo de sub-redes.
 - b. Na caixa Description (Descrição), digite uma descrição para seu grupo de sub-redes.
 - c. Na caixa VPC ID (ID da VPC_), escolha a Amazon VPC que você criou.
 - d. Nas listas Zona de disponibilidade e ID de sub-rede, escolha a zona de disponibilidade ou [Uso de zonas locais com o ElastiCache](#) e o ID da sub-rede privada e depois escolha Adicionar.

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

VPC ID

The identifier for the VPC environment where your cluster is to run.

 [Create VPC](#)

i For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6) [Manage](#)

Availability Zone ▲	Subnet ID ▼	Outpost ID ▼	CIDR block ▼
us-east-1a	subnet-		172.31.16.0/20
us-east-1b	subnet-		172.31.32.0/20
us-east-1c	subnet-		172.31.0.0/20
us-east-1d	subnet-		172.31.80.0/20

5. Na mensagem de confirmação exibida, escolha Close (Fechar).

Seu novo grupo de sub-redes aparece na lista Grupos de sub-redes do ElastiCache console. Na parte inferior da janela, você pode escolher o grupo de sub-redes para ver detalhes, como todas as sub-redes associadas a esse grupo.

Criar um grupo de sub-redes (AWS CLI)

No prompt de comando, use o comando `create-cache-subnet-group` para criar um grupo de sub-redes.

Para Linux, macOS ou Unix:

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

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

Para Windows:

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

Esse comando deve produzir um resultado semelhante ao seguinte:

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

Para obter mais informações, consulte o AWS CLI tópico [create-cache-subnet-group](#).

Etapa 2: criar um cluster

Antes de criar um cluster para uso em produção, é óbvio que você precisa considerar como configurar o cluster para atender às suas necessidades de negócios. Esses problemas são abordados na seção [Preparação de um cluster no ElastiCache](#). Para os propósitos deste exercício de introdução, você criará um cluster com o modo cluster desabilitado e poderá aceitar os valores de configuração padrão onde eles se aplicam.

O cluster que você criará estará ativo, e não em execução em uma sandbox. Você incorrerá nas taxas ElastiCache de uso padrão da instância até excluí-la. As cobranças totais serão mínimas (geralmente menos de um dólar) se você concluir o exercício descrito aqui em uma única sessão e

excluir seu cluster quando terminar. Para obter mais informações sobre taxas de ElastiCache uso, consulte [Amazon ElastiCache](#).

Seu cluster é iniciado em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC.

Criação de um cluster do Valkey (modo cluster desabilitado) (console)

Para criar um cluster Valkey (modo de cluster desativado) usando o console ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região na qual você deseja iniciar esse cluster.
3. No painel de navegação, escolha Get started (Conceitos básicos).
4. Escolha Create VPC (Criar VPC) e siga as etapas em [Creating a Virtual Private Cloud \(VPC\)](#) (Criação de uma nuvem privada virtual (VPC)).
5. Na página do ElastiCache painel, escolha Valkey cache ou Redis OSS cache e, em seguida, escolha Create Valkey cache ou Create Redis OSS cache.
6. Em Cluster settings (Configurações do cluster), faça o seguinte:
 - a. Selecione Configure and create a new cluster (Configurar e criar um novo cluster).
 - b. Em Cluster mode (Modo cluster), escolha Disabled (Desabilitado).
 - c. Para o Cluster info (Informações sobre o cluster), insira um valor para Name (Nome).
 - d. (Opcional) Insira um valor para Description (Descrição).
7. Em Location (Local):

AWS Cloud

1. Para AWS Cloud (Nuvem), recomendamos que você aceite as configurações padrão de Multi-AZ e Failover automático. Para obter mais informações, consulte [Minimizando o tempo de inatividade no ElastiCache Redis OSS](#) com Multi-AZ.
2. Em Cluster settings (Configurações do cluster)
 - a. Em Engine version (Versão do mecanismo), escolha a versão do mecanismo.
 - b. Em Port (Porta), use a porta padrão, 6379. Se você tiver um motivo para usar uma porta diferente, insira o número da porta.
 - c. Em Grupo de parâmetro (Parameter group), escolha um grupo de parâmetro ou crie um novo. Os grupo de parâmetros controlam os parâmetros de tempo de execução

do seu cluster. Para obter mais informações sobre grupo de parâmetros, consulte [Parâmetros do Valkey e do Redis OSS](#) e [Criação de um grupo de parâmetros do ElastiCache](#).

 Note

Quando você seleciona um grupo de parâmetros para definir os valores de configuração do mecanismo, esse grupo de parâmetros é aplicado a todos os clusters no datastore global. Na página Parameter Groups (Grupos de parâmetros), o atributo Global indica se um grupo de parâmetros faz parte de um datastore global.

- d. Para Node type (Tipo de nó), escolha a seta para baixo (▼).

Na caixa de diálogo Change node type (Alterar tipo de nó), escolha um valor para a Instance family (Família de instâncias) para o tipo de nó desejado. Depois disso, escolha o tipo de nó que deseja usar para este cluster e escolha Save (Salvar).

Para obter mais informações, consulte [Escolha do tamanho do nó](#).

Se você escolher um tipo de nó r6gd, a classificação de dados em níveis será ativada automaticamente. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

- e. Em Number of nodes (Número de nós), escolha o número de réplicas de leitura desejado. Se você tiver habilitado Multi-AZ, o número deverá estar entre 1 e 5.
3. Em Connectivity (Conectividade)
- a. Em Network type (Tipo de rede), escolha as versões de IP compatíveis com esse cluster.
- b. Em Grupos de sub-redes, escolha a sub-rede que você deseja aplicar a esse cluster. ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. ElastiCache os clusters exigem uma sub-rede de pilha dupla com ambos IPv4 e IPv6 endereços atribuídos a eles para operar no modo de pilha dupla e uma IPv6 sub-rede somente para operar somente como -only. IPv6

Ao criar um novo grupo de sub-rede, insira o VPC ID (ID da VPC) ao qual ele pertence.

Para obter mais informações, consulte:

- [Escolhendo um tipo de rede em ElastiCache](#).
- [Criar uma sub-rede na VPC](#).

Se você for [Uso de zonas locais com o ElastiCache](#), deverá criar ou escolher uma sub-rede que esteja na região local.

Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).

4. Em Availability zone placements (Posicionamentos de zona de disponibilidade), você tem duas opções:
 - Sem preferência — ElastiCache escolhe a Zona de Disponibilidade.
 - Specify availability zones (Especificar zonas de disponibilidade): você especifica a zona de disponibilidade para cada cluster.

Se optar por especificar as zonas de disponibilidade, para cada cluster em cada fragmento, escolha a zona de disponibilidade na lista.

Para obter mais informações, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#).

5. Escolha Next (Próximo)
6. Em Configurações avançadas do Valkey ou do Redis OSS
 - Para Security (Segurança):
 - i. Para criptografar seus dados, você tem as seguintes opções:
 - Criptografia em repouso: permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

 Note

Você tem a opção de fornecer uma chave de criptografia diferente escolhendo a chave AWS KMS gerenciada pelo cliente e escolhendo

a chave. Para obter mais informações, consulte [Usar chaves gerenciadas pelo cliente do AWS KMS](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [criptografia em trânsito](#). Para o Valkey ou Redis OSS versão 6.0 em diante, ao habilitar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Access Control (Controle de acesso):
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.
 - User Group Access Control List (Lista de controle de acesso de grupo de usuários): selecione um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para obter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
 - AUTH Default User (Usuário padrão AUTH): um mecanismo de autenticação para o servidor Redis OSS. Para obter mais informações, consulte [AUTH](#).
- AUTH: um mecanismo de autenticação para o servidor Redis OSS. Para obter mais informações, consulte [AUTH](#).

 Note

Para versões do Valkey e do Redis OSS 3.2.6 em diante, excluindo a versão 3.2.10, o Redis OSS AUTH é a única opção.

- ii. Em Grupos de segurança, escolha os grupos de segurança desejados para esse cluster. Um grupo de segurança atua como um firewall para controlar o acesso à rede ao cluster. É possível usar o grupo de segurança padrão para sua VPC ou criar um novo.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança para sua VPC](#) no Guia do usuário da Amazon VPC.
7. Para backups automáticos agendados regularmente, selecione Enable automatic backups (Habilitar backups automáticos) e insira o número de dias que deseja que cada backup automático seja mantido antes de ser excluído automaticamente. Se não quiser backups automáticos agendados regularmente, desmarque a caixa de seleção Enable

automatic backups (Habilitar backups automáticos). Em ambos os casos, você sempre tem a opção de criar backups manuais.

Para obter mais informações sobre o backup e a restauração do Redis OSS, consulte [Snapshots e restauração](#).

8. (Opcional) Especifique uma janela de manutenção. A opção janela de manutenção é o tempo, geralmente uma hora de duração, a cada semana quando o ElastiCache agenda a manutenção do sistema para seu cluster. Você pode permitir que o ElastiCache escolha o dia e a hora da sua janela de manutenção (No preference) ou pode escolher o dia, a hora e a duração por conta própria (Specify maintenance window). Se você escolher Especificar janela de manutenção, nas listas, escolha Dia de início, Hora de início e Duração (em horas) para sua janela de manutenção. Todos os horários são em UCT.

Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

9. (Opcional) Para Logs:
 - Em Log format (Formato do log), escolha Text (Texto) ou JSON.
 - Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
 - Em Destino do registro, escolha Criar novo e insira o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose, ou escolha Selecionar existente e escolha o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose,
10. Para Tags, para ajudá-lo a gerenciar seus clusters e outros ElastiCache recursos, você pode atribuir seus próprios metadados a cada recurso na forma de tags. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).
11. Escolha Next (Próximo).
12. Revise todas as suas entradas e opções e faça as correções necessárias. Quando estiver pronto, escolha Create (Criar).

On premises

1. Para On-premises, recomendamos que você deixe Auto-failover (Failover automático) habilitado. Para obter mais informações, consulte [Minimizando o tempo de inatividade no ElastiCache Redis](#) OSS com Multi-AZ

2. Para concluir a criação do cluster, siga as etapas em [Using Outposts](#) (Uso do Outposts).

Assim que o status do seu cluster estiver disponível, você poderá conceder EC2 à Amazon acesso a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4: Conectar-se ao nó de um cluster](#).

 Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Para trabalhar com o modo de cluster habilitado, consulte os seguintes tópicos:

- Para usar o console, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#).
- Para usar o AWS CLI, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(AWS CLI\)](#).

Etapa 3. Autorizar o acesso ao cluster

Esta seção pressupõe que você esteja familiarizado com o lançamento e a conexão com EC2 instâncias da Amazon. Para obter mais informações, consulte o [Amazon EC2 Getting Started Guide](#).

Todos os ElastiCache clusters são projetados para serem acessados a partir de uma EC2 instância da Amazon. O cenário mais comum é acessar um ElastiCache cluster de uma EC2 instância da Amazon na mesma Amazon Virtual Private Cloud (Amazon VPC), o que será o caso deste exercício.

Por padrão, o acesso de rede ao seu cluster é limitado à conta que foi usada para criá-lo. Antes de se conectar a um cluster a partir de uma EC2 instância, você deve autorizar a EC2 instância a acessar o cluster.

O caso de uso mais comum é quando um aplicativo implantado em uma EC2 instância precisa se conectar a um cluster na mesma VPC. A maneira mais simples de gerenciar o acesso entre EC2 instâncias e clusters na mesma VPC é fazer o seguinte:

1. Crie um grupo de segurança de VPC para o seu cluster. Esse security group pode ser usado para restringir o acesso às instâncias do cluster. Por exemplo, é possível criar uma regra personalizada para esse grupo de segurança que permite o acesso TCP usando a porta atribuída ao cluster quando você o criou e um endereço IP que será usado para acessar o cluster.

A porta padrão dos clusters e grupos de replicação Valkey ou Redis OSS é 6379.

Important

Os grupos ElastiCache de segurança da Amazon são aplicáveis somente a clusters que não estão sendo executados em um ambiente Amazon Virtual Private Cloud (VPC). Se você estiver executando em uma Amazon Virtual Private Cloud, os Grupos de segurança não estarão disponíveis no painel de navegação do console.

Se você estiver executando seus ElastiCache nós em uma Amazon VPC, você controla o acesso aos seus clusters com grupos de segurança da Amazon VPC, que são diferentes dos grupos de segurança. ElastiCache Para obter mais informações sobre o uso ElastiCache em uma Amazon VPC, consulte [Amazon VPCs e ElastiCache segurança](#)

2. Crie um grupo de segurança VPC para suas EC2 instâncias (servidores web e de aplicativos). Esse grupo de segurança pode, se necessário, permitir o acesso à EC2 instância pela Internet

por meio da tabela de roteamento da VPC. Por exemplo, você pode definir regras nesse grupo de segurança para permitir acesso TCP à EC2 instância pela porta 22.

3. Crie regras personalizadas no grupo de segurança do seu cluster que permitam conexões do grupo de segurança que você criou para suas EC2 instâncias. Isso permitiria que qualquer membro de grupo de segurança acessasse os clusters.

 Note

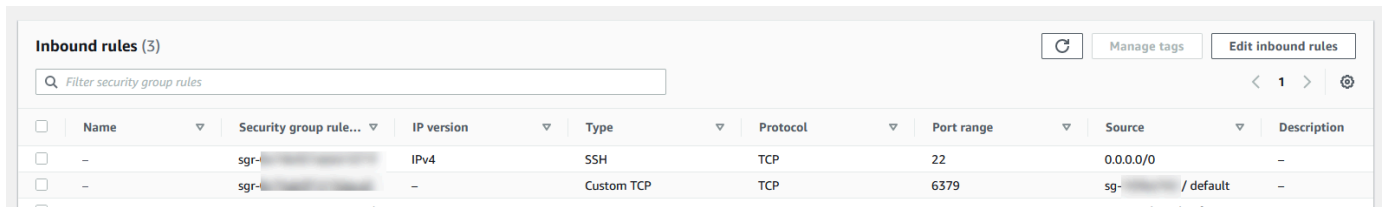
Se você estiver planejando usar [Zonas locais](#), verifique se você as habilitou. Quando você criar um grupo de sub-redes nessa região local, sua VPC será estendida para essa zona local e sua VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade. Todos os gateways e tabelas de rotas relevantes serão ajustados automaticamente.

Para criar uma regra em um grupo de segurança de VPC que permita conexões de outro grupo de segurança

1. [Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.](https://console.aws.amazon.com/vpc)
2. No painel de navegação, escolha Security Groups (Grupos de segurança).
3. Selecione ou crie um security group que você usará para suas instâncias de cluster. Em Regras de entrada, selecione Editar regras de entrada e escolha Adicionar regra. Esse grupo de segurança permitirá o acesso a membros de outro grupo de segurança.
4. Em Tipo, escolha Regra TCP personalizada.
 - a. Para Port Range, especifique a porta que você usou quando criou seu cluster.

A porta padrão dos clusters e grupos de replicação Valkey ou Redis OSS é 6379.

- b. Na caixa Source, comece a digitar o ID do grupo de segurança. Na lista, selecione o grupo de segurança que você usará para suas EC2 instâncias da Amazon.
5. Escolha Save quando terminar.



	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
☐	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
☐	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-
☐	-	-	-	-	-	-	-	-

Depois de habilitar o acesso, você agora estará pronto para se conectar ao nó, conforme discutido na próxima seção.

Para obter informações sobre como acessar seu ElastiCache cluster de uma Amazon VPC diferente, de uma AWS região diferente ou até mesmo de sua rede corporativa, consulte o seguinte:

- [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#)
- [Acessando ElastiCache recursos de fora AWS](#)

Etapa 4: Conectar-se ao nó de um cluster

Antes de continuar, conclua [Etapa 3. Autorizar o acesso ao cluster](#).

Esta seção pressupõe que você criou uma EC2 instância da Amazon e pode se conectar a ela. Para obter instruções sobre como fazer isso, consulte o [Amazon EC2 Getting Started Guide](#).

Uma EC2 instância da Amazon pode se conectar a um nó de cluster somente se você a tiver autorizado a fazer isso.

Localize seus endpoints de nó

Quando seu cluster está no estado disponível e você autorizou o acesso a ele, você pode fazer login em uma EC2 instância da Amazon e se conectar ao cluster. Para isso, primeiro você deve determinar o endpoint.

Localização de endpoints de um cluster do Valkey (modo cluster desabilitado) (console)

Se o cluster do Valkey (modo cluster desabilitado) tiver apenas um nó, o endpoint desse nó será usado para tanto para leituras quanto para gravações. Se cluster tiver vários nós, haverá três tipos de endpoints: o endpoint primário, o endpoint leitor e os endpoints de nó.

O endpoint primário é um nome DNS que sempre é resolvido para o nó primário no cluster. O endpoint primário é imune às alterações no seu cluster, como promover uma réplica de leitura para a função primária. Para atividades de gravação, recomendamos que suas aplicações se conectem ao endpoint primário.

Um endpoint de leitura dividirá uniformemente as conexões de entrada com o endpoint entre todas as réplicas de leitura em um cluster. ElastiCache Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar as várias réplicas de leitura do seu ElastiCache cluster em diferentes zonas de AWS disponibilidade (AZ) para garantir a alta disponibilidade dos endpoints de leitura.

Note

Um endpoint leitor não é um balanceador de carga. É um registro DNS que vai resolver para um endereço IP de um dos nós de réplica em uma forma round robin.

Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó no seu aplicativo.

Para localizar um endpoint de um cluster do Valkey (modo cluster desabilitado) (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Caches do Valkey ou Caches do Redis OSS.

A tela de clusters aparecerá com uma lista que vai incluir todos os caches com tecnologia sem servidor do Valkey ou Redis OSS, cluster do Valkey (modo de cluster desabilitado) e cluster do Valkey (modo de cluster habilitado). Selecione o cluster que você criou na seção [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#).

3. Para encontrar os endpoints do Primary and/or Reader do cluster, escolha o nome do cluster (não o botão de opção).

▼ Cluster details			
Cluster name [redacted]	Description [redacted]	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Endpoints primários e leitores para um cluster do Valkey (modo cluster desabilitado)

Se houver apenas um nó no cluster, não haverá um endpoint primário, e você poderá continuar na próxima etapa.

4. Se o cluster do Valkey (modo cluster desabilitado) tiver nós de réplica, você poderá encontrar seus endpoints dos nós de réplica escolhendo o nome do cluster e escolhendo a guia Nodes (Nós).

A tela de nós aparece com cada nó do cluster, primário e réplicas, listados com seu respectivo endpoint.

☐	Node Name	Status	Current Role	Port	Endpoint
☐	test-no-001	available	primary	6379	test-no-001.usw2.cache.amazonaws.com:6379
☐	test-no-002	available	replica	6379	test-no-002.usw2.cache.amazonaws.com:6379
☐	test-no-003	available	replica	6379	test-no-003.usw2.cache.amazonaws.com:6379

Endpoints de nó para um cluster do Valkey (modo cluster desabilitado)

5. Como copiar um endpoint para a área de transferência:

- Um endpoint por vez, localize o endpoint que você deseja copiar.
- Escolha o ícone de cópia diretamente na frente do endpoint.

O endpoint agora é copiado para a área de transferência. Para obter informações sobre como usar o endpoint para se conectar a um nó, consulte [Conexão a nós](#).

Um endpoint primário do Valkey (modo cluster desabilitado) se assemelha ao seguinte. Há uma diferença, dependendo de a criptografia em trânsito estar ou não habilitada.

Criptografia em trânsito não habilitada

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

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

Criptografia em trânsito habilitada

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

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

Para explorar mais sobre como localizar os endpoints, consulte os tópicos relevantes para o tipo de mecanismo e de cluster que você está executando.

- [Encontrar endpoints de conexão no ElastiCache](#)

- [Localização de endpoints para um cluster do Valkey ou do Redis OSS \(modo cluster habilitado\) \(console\)](#): você precisa do endpoint de configuração do cluster.
- [Localizar endpoints \(AWS CLI\)](#)
- [Localização de endpoints \(API do ElastiCache\)](#)

Conexão a um cluster ou grupo de replicação do Valkey ou Redis OSS (Linux)

Agora que você tem o endpoint de que precisa, pode fazer login em uma EC2 instância e se conectar ao cluster ou ao grupo de replicação. No exemplo a seguir, você usa o utilitário `valkey-cli` para se conectar a um cluster. A versão mais recente do `valkey-cli` também suporta SSL/TLS a conexão de clusters habilitados. `encryption/authentication`

O exemplo a seguir usa EC2 instâncias da Amazon executando o Amazon Linux e o Amazon Linux 2. Para obter detalhes sobre como instalar e compilar a `valkey-cli` com outras distribuições Linux, consulte a documentação do seu sistema operacional específico.

 Note

Este processo abrange o teste de uma conexão usando o utilitário `valkey-cli` somente para uso não planejado. Para obter uma lista dos clientes de Valkey e Redis OSS compatíveis, consulte a [Documentação do Valkey](#). Para obter exemplos de uso do AWS SDKs com ElastiCache, consulte [Tutoriais: conceitos básicos do Python e do ElastiCache](#).

Conexão a um cluster não criptografado com modo cluster desabilitado

1. Execute o comando a seguir para se conectar ao cluster *primary-endpoint* e substituí-lo *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Valkey ou Redis OSS é 6379.)

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

O resultado em um prompt de comando do Valkey ou Redis OSS é semelhante ao seguinte:

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

2. Agora você pode executar comandos do Valkey ou Redis OSS.

```
set x Hello
OK

get x
"Hello"
```

Conectar a um cluster não criptografado com modo cluster habilitado

1. Execute o comando a seguir para se conectar ao cluster *configuration-endpoint* e substituí-lo *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Valkey ou Redis OSS é 6379.)

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

 Note

No comando anterior, a opção -c habilita o modo cluster em seguida aos [redirecionamentos -ASK e -MOVED](#).

O resultado em um prompt de comando do Valkey ou Redis OSS é semelhante ao seguinte:

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

2. Agora você pode executar comandos do Valkey ou Redis OSS. Observe que o redirecionamento ocorre porque você o habilitou usando a opção -c. Se o redirecionamento não estiver habilitado, o comando retornará o erro MOVED. Para obter mais informações sobre o erro MOVED, consulte [Especificação do cluster do Redis OSS](#).

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

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

Conectando-se a um cluster Encryption/Authentication habilitado

Por padrão, a valkey-cli usa uma conexão TCP não criptografada ao se conectar ao Valkey ou Redis OSS. A opção é BUILD_TLS=yes SSL/TLS ativada no momento da compilação do valkey-cli, conforme mostrado na seção anterior. [Baixar e configurar o acesso à linha de comando](#) A ativação do AUTH é opcional. No entanto, você deve habilitar a criptografia em trânsito para habilitar o AUTH. Para obter mais detalhes sobre ElastiCache criptografia e autenticação, consulte [ElastiCache criptografia em trânsito \(TLS\)](#).

Note

Você pode usar a opção `--tls` com a valkey-cli para se conectar a clusters criptografados com modo cluster habilitado e desabilitado. Se um cluster tiver um conjunto de tokens AUTH, você poderá usar a opção `-a` para fornecer uma senha do AUTH.

Nos exemplos a seguir, certifique-se de substituir *cluster-endpoint* e *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Valkey ou Redis OSS é 6379.)

Conexão a clusters criptografados com modo cluster desabilitado

O exemplo a seguir conecta a um cluster habilitado para criptografia e autenticação:

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

O exemplo a seguir conecta a um cluster somente habilitado para criptografia:

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

Conexão a clusters criptografados com modo cluster habilitado

O exemplo a seguir conecta a um cluster habilitado para criptografia e autenticação:

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

O exemplo a seguir conecta a um cluster somente habilitado para criptografia:

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

Depois de se conectar ao cluster, você poderá executar os comandos do Valkey ou Redis OSS, conforme mostrado nos exemplos anteriores para clusters não criptografados.

Alternativa com valkey-cli

Se o cluster não estiver habilitado para o modo cluster e você precisar fazer uma conexão com o cluster para um teste curto, mas sem passar pela compilação da valkey-cli, você pode usar telnet ou openssl. Nos comandos de exemplo a seguir, certifique-se de substituir *cluster-endpoint* e *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Valkey ou Redis OSS é 6379.)

O exemplo a seguir se conecta a um cluster desativado no modo de cluster habilitado para and/or autenticação por criptografia:

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

Se o cluster tiver uma senha definida, conecte-se primeiro ao cluster. Após a conexão, autentique o cluster usando o seguinte comando, depois pressione a tecla Enter. No exemplo a seguir, *your-password* substitua pela senha do seu cluster.

```
Auth your-password
```

O exemplo a seguir conecta a um cluster com modo cluster desabilitado que não tenha criptografia ou autenticação habilitada:

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

Conexão a um cluster ou grupo de replicação do Valkey ou Redis OSS (Windows)

Para se conectar ao cluster Valkey ou Redis OSS a partir de uma instância do EC2 Windows usando a CLI Valkey ou a CLI do Redis OSS, você deve baixar o pacote valkey-cli e usar o valkey-cli.exe para se conectar ao cluster Valkey ou Redis OSS a partir de uma instância do Windows. EC2

No exemplo a seguir, use o utilitário `valkey-cli` para se conectar a um cluster que não está habilitado para criptografia e que está executando o Valkey ou Redis OSS. Para obter mais informações sobre o Valkey ou Redis OSS e os comandos disponíveis, consulte [Comandos do Valkey ou Redis OSS](#) no site do Valkey.

Para se conectar a um cluster do Valkey ou Redis OSS não habilitado para criptografia usando o `valkey-cli`

1. Conecte-se à sua EC2 instância da Amazon usando o utilitário de conexão de sua escolha. Para obter instruções sobre como se conectar a uma EC2 instância da Amazon, consulte o [Amazon EC2 Getting Started Guide](#).
2. Copie e cole o link <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> em um navegador da Internet para baixar o arquivo zip para o cliente Valkey a partir da versão disponível em GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extraia o arquivo zip para a pasta/caminho desejado.

Abra o prompt de comando, mude para o diretório do Valkey e execute o comando `c :`
`\Valkey>valkey-cli -h Redis_Cluster_Endpoint -p 6379`.

Por exemplo:

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

3. Execute os comandos Valkey ou Redis OSS.

Agora, você está conectado ao cluster e pode executar comandos Valkey ou Redis OSS como os seguintes.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5   // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                   // Get value for key "b"
"Good-bye"
                        // wait >= 5 seconds
```



```
get b
(nil)           // key has expired, nothing returned
quit           // Exit from valkey-cli
```

O que faço agora?

Agora que você experimentou o exercício de introdução, pode explorar as seções a seguir para saber mais sobre ElastiCache as ferramentas disponíveis:

- [Começando com AWS](#)
- [Ferramentas para a Amazon Web Services](#)
- [AWS Command Line Interface](#)
- [Referência de ElastiCache API da Amazon](#)

Depois de concluir o exercício de introdução, você pode ler estas seções para saber mais sobre ElastiCache administração:

- [Escolha do tamanho do nó](#)

Você quer que seu cache seja grande o suficiente para acomodar todos os dados que deseja armazenar em cache. Ao mesmo tempo, você não quer pagar mais cache do que precisa. Use este tópico para ajudá-lo a escolher o melhor tamanho de nó.

- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)

Identifique e lide com os problemas que podem afetar a eficiência de seu cluster.

Criação de um ElastiCache cluster baseado em nós para o Redis OSS

A seguir estão as ações únicas que você deve realizar para criar um ElastiCache cluster baseado em nós para o Redis OSS.

Para obter mais informações sobre a configuração, ElastiCache consulte [Conf ElastiCacheiguração](#).

Tópicos

- [Etapa 1: criar um grupo de sub-redes](#)
- [Etapa 2: criar um cluster](#)

- [Etapa 3: autorizar o acesso ao cluster](#)
- [Etapa 4: conexão ao nó do cluster](#)

Etapa 1: criar um grupo de sub-redes

Antes de criar seu cluster, crie primeiro um grupo de sub-redes. Um grupo de sub-redes de cache é uma coleção de sub-redes que você pode querer designar aos seus clusters em uma VPC. Ao executar um cluster em uma VPC, é necessário selecionar um grupo de sub-redes de cache. Em seguida, ElastiCache usa esse grupo de sub-redes de cache para atribuir endereços IP dentro dessa sub-rede a cada nó de cache no cluster.

Quando você criar um novo grupo de sub-redes, observe o número de endereços IP disponíveis. Se a sub-rede tiver muito poucos endereços IP livres, talvez haja um limite no que diz respeito ao número de nós adicionais que é possível acrescentar ao cluster. Para resolver esse problema, você pode atribuir uma ou mais sub-redes a um grupo de sub-redes para ter um número suficiente de endereços IP na zona de disponibilidade do seu cluster. Depois disso, você pode adicionar mais nós ao seu cluster.

Os procedimentos a seguir mostram como criar um grupo de sub-rede chamado mysubnetgroup (console) e a AWS CLI.

Criação de um grupo de sub-redes (console)

O procedimento a seguir mostra como criar um grupo de sub-redes (console).

Como criar um grupo de sub-redes (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha Subnet Groups (Grupos de sub-redes).
3. Selecione Create Subnet Group (Criar grupo de sub-redes).
4. No assistente Create Subnet Group (Criar grupo de sub-redes), faça o seguinte. Quando estiver satisfeito com todas as configurações, clique em Yes, Create (Sim, criar).
 - a. Na caixa Name (Nome), digite um nome para o seu grupo de sub-redes.
 - b. Na caixa Description (Descrição), digite uma descrição para seu grupo de sub-redes.
 - c. Na caixa VPC ID (ID da VPC_), escolha a Amazon VPC que você criou.

- d. Nas listas Availability Zone (Zona de disponibilidade) e Subnet ID (ID da sub-rede), escolha a zona de disponibilidade ou a [Local Zone](#) (Zona local) e o ID da sub-rede privada, e depois escolha Add (Adicionar).

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

my-subnet-group

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

test

VPC ID

The identifier for the VPC environment where your cluster is to run.

vpc-

Create VPC

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6)

Manage

Availability Zone	Subnet ID	Outpost ID	CIDR block
us-east-1a	subnet-		172.31.16.0/20
us-east-1b	subnet-		172.31.32.0/20
us-east-1c	subnet-		172.31.0.0/20
us-east-1d	subnet-		172.31.80.0/20

5. Na mensagem de confirmação exibida, escolha Close (Fechar).

Seu novo grupo de sub-redes aparece na lista Grupos de sub-redes do ElastiCache console. Na parte inferior da janela, você pode escolher o grupo de sub-redes para ver detalhes, como todas as sub-redes associadas a esse grupo.

Criar um grupo de sub-redes (AWS CLI)

No prompt de comando, use o comando `create-cache-subnet-group` para criar um grupo de sub-redes.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Esse comando deve produzir um resultado semelhante ao seguinte:

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

Para obter mais informações, consulte o AWS CLI tópico [create-cache-subnet-group](#).

Etapa 2: criar um cluster

Antes de criar um cluster para uso em produção, é óbvio que você precisa considerar como configurar o cluster para atender às suas necessidades de negócios. Esses problemas são abordados na seção [Preparação de um cluster no ElastiCache](#). Para os propósitos deste exercício de introdução, você criará um cluster com o modo cluster desabilitado e poderá aceitar os valores de configuração padrão onde eles se aplicam.

O cluster que você criará estará ativo, e não em execução em uma sandbox. Você incorrerá nas taxas ElastiCache de uso padrão da instância até excluí-la. As cobranças totais serão mínimas (geralmente menos de um dólar) se você concluir o exercício descrito aqui em uma única sessão e excluir seu cluster quando terminar. Para obter mais informações sobre taxas de ElastiCache uso, consulte [Amazon ElastiCache](#).

Seu cluster é iniciado em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC.

Criação de um cluster do Redis OSS (modo cluster desabilitado) (console)

Para criar um cluster Redis OSS (modo de cluster desativado) usando o console ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região na qual você deseja iniciar esse cluster.
3. No painel de navegação, escolha Get started (Conceitos básicos).
4. Escolha Create VPC (Criar VPC) e siga as etapas em [Creating a Virtual Private Cloud \(VPC\)](#) (Criação de uma nuvem privada virtual (VPC)).
5. Na página do ElastiCache painel, escolha Valkey cache ou Redis OSS cache. Para este exercício, escolheremos o cache Redis OSS e, em seguida, Criar cache Redis OSS.
6. Em Cluster settings (Configurações do cluster), faça o seguinte:
 - a. Selecione Configure and create a new cluster (Configurar e criar um novo cluster).
 - b. Em Cluster mode (Modo cluster), escolha Disabled (Desabilitado).
 - c. Para o Cluster info (Informações sobre o cluster), insira um valor para Name (Nome).
 - d. (Opcional) Insira um valor para Description (Descrição).
7. Em Location (Local):

AWS Cloud

1. Para AWS Cloud (Nuvem), recomendamos que você aceite as configurações padrão de Multi-AZ e Failover automático. Para obter mais informações, consulte [Minimizando o tempo de inatividade no ElastiCache Redis OSS](#) com Multi-AZ.
2. Em Cluster settings (Configurações do cluster)
 - a. Em Engine version (Versão do mecanismo), escolha a versão do mecanismo.

- b. Em Port (Porta), use a porta padrão, 6379. Se você tiver um motivo para usar uma porta diferente, insira o número da porta.
- c. Em Grupo de parâmetro (Parameter group), escolha um grupo de parâmetro ou crie um novo. Os grupo de parâmetros controlam os parâmetros de tempo de execução do seu cluster. Para obter mais informações sobre grupo de parâmetros, consulte [Parâmetros do Valkey e do Redis OSS](#) e [Criação de um grupo de parâmetros do ElastiCache](#).

 Note

Quando você seleciona um grupo de parâmetros para definir os valores de configuração do mecanismo, esse grupo de parâmetros é aplicado a todos os clusters no datastore global. Na página Parameter Groups (Grupos de parâmetros), o atributo Global indica se um grupo de parâmetros faz parte de um datastore global.

- d. Para Node type (Tipo de nó), escolha a seta para baixo (▼)).
Na caixa de diálogo Change node type (Alterar tipo de nó), escolha um valor para a Instance family (Família de instâncias) para o tipo de nó desejado. Depois disso, escolha o tipo de nó que deseja usar para este cluster e escolha Save (Salvar).

Para obter mais informações, consulte [Escolha do tamanho do nó](#).

Se você escolher um tipo de nó r6gd, a classificação de dados em níveis será ativada automaticamente. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

- e. Em Number of nodes (Número de nós), escolha o número de réplicas de leitura desejado. Se você tiver habilitado Multi-AZ, o número deverá estar entre 1 e 5.
3. Em Connectivity (Conectividade)
- a. Em Network type (Tipo de rede), escolha as versões de IP compatíveis com esse cluster.
 - b. Em Grupos de sub-redes, escolha a sub-rede que você deseja aplicar a esse cluster. ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. ElastiCache os clusters exigem uma sub-rede de pilha dupla com ambos IPv4 e IPv6 endereços

atribuídos a eles para operar no modo de pilha dupla e uma IPv6 sub-rede somente para operar como somente. IPv6

Ao criar um novo grupo de sub-rede, insira o VPC ID (ID da VPC) ao qual ele pertence.

Para obter mais informações, consulte:

- [Escolhendo um tipo de rede em ElastiCache.](#)
- [Criar uma sub-rede na VPC.](#)

Se você for [Uso de zonas locais com o ElastiCache](#), deverá criar ou escolher uma sub-rede que esteja na região local.

Para obter mais informações, consulte [Sub-redes e grupos de sub-redes.](#)

4. Em Availability zone placements (Posicionamentos de zona de disponibilidade), você tem duas opções:
 - Sem preferência — ElastiCache escolhe a Zona de Disponibilidade.
 - Specify availability zones (Especificar zonas de disponibilidade): você especifica a zona de disponibilidade para cada cluster.

Se optar por especificar as zonas de disponibilidade, para cada cluster em cada fragmento, escolha a zona de disponibilidade na lista.

Para obter mais informações, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache.](#)

5. Escolha Next (Próximo)
6. Em Advanced Redis OSS settings
 - Para Security (Segurança):
 - i. Para criptografar seus dados, você tem as seguintes opções:
 - Criptografia em repouso: permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso.](#)

 Note

Você tem a opção de fornecer uma chave de criptografia diferente escolhendo a chave AWS KMS gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usar chaves gerenciadas pelo cliente do AWS KMS](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [criptografia em trânsito](#). Para a versão 6.0 ElastiCache do mecanismo para Redis OSS e versões posteriores, se você ativar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de controle de acesso:
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.
 - User Group Access Control List (Lista de controle de acesso de grupo de usuários): selecione um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para obter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
 - AUTH Default User (Usuário padrão AUTH): um mecanismo de autenticação para o servidor Valkey e Redis OSS. Para obter mais informações, consulte [AUTH](#).
- AUTH: um mecanismo de autenticação para o servidor Redis OSS. Para obter mais informações, consulte [AUTH](#).

 Note

Para versões do Redis OSS 3.2.6 em diante, excluindo a versão 3.2.10, o Redis OSS AUTH é a única opção.

- ii. Em Grupos de segurança, escolha os grupos de segurança desejados para esse cluster. Um grupo de segurança atua como um firewall para controlar o acesso à rede ao cluster. É possível usar o grupo de segurança padrão para sua VPC ou criar um novo.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança para sua VPC](#) no Guia do usuário da Amazon VPC.

7. Para backups automáticos agendados regularmente, selecione Enable automatic backups (Habilitar backups automáticos) e insira o número de dias que deseja que cada backup automático seja mantido antes de ser excluído automaticamente. Se não quiser backups automáticos agendados regularmente, desmarque a caixa de seleção Enable automatic backups (Habilitar backups automáticos). Em ambos os casos, você sempre tem a opção de criar backups manuais.

Para obter mais informações sobre backup e restauração, consulte [Snapshots e restauração](#).

8. (Opcional) Especifique uma janela de manutenção. A opção janela de manutenção é o tempo, geralmente uma hora de duração, a cada semana quando o ElastiCache agenda a manutenção do sistema para seu cluster. Você pode permitir que o ElastiCache escolha o dia e a hora da sua janela de manutenção (No preference) ou pode escolher o dia, a hora e a duração por conta própria (Specify maintenance window). Se você escolher Especificar janela de manutenção, nas listas, escolha Dia de início, Hora de início e Duração (em horas) para sua janela de manutenção. Todos os horários são em UCT.

Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

9. (Opcional) Para Logs:
 - Em Log format (Formato do log), escolha Text (Texto) ou JSON.
 - Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
 - Em Destino do registro, escolha Criar novo e insira o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose, ou escolha Selecionar existente e escolha o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose,
10. Para Tags, para ajudá-lo a gerenciar seus clusters e outros ElastiCache recursos, você pode atribuir seus próprios metadados a cada recurso na forma de tags. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).
11. Escolha Next (Próximo).

12. Revise todas as suas entradas e opções e faça as correções necessárias. Quando estiver pronto, escolha Create (Criar).

On premises

1. Para On-premises, recomendamos que você deixe Auto-failover (Failover automático) habilitado. Para obter mais informações, consulte [Minimizando o tempo de inatividade no ElastiCache Redis](#) OSS com Multi-AZ
2. Para concluir a criação do cluster, siga as etapas em [Using Outposts](#) (Uso do Outposts).

Assim que o status do seu cluster estiver disponível, você poderá conceder EC2 à Amazon acesso a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4: Conectar-se ao nó de um cluster](#).

Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Criação de um cluster do Redis OSS (modo de cluster desabilitado) (AWS CLI)

Example

O código da CLI a seguir cria um cluster Redis OSS (modo cluster desabilitado) sem réplicas.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Para trabalhar com o modo de cluster habilitado, consulte os seguintes tópicos:

- Para usar o console, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#).
- Para usar o AWS CLI, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(AWS CLI\)](#).

Etapa 3: autorizar o acesso ao cluster

Esta seção pressupõe que você esteja familiarizado com o lançamento e a conexão com EC2 instâncias da Amazon. Para obter mais informações, consulte o [Amazon EC2 Getting Started Guide](#).

Todos os ElastiCache clusters são projetados para serem acessados a partir de uma EC2 instância da Amazon. O cenário mais comum é acessar um ElastiCache cluster de uma EC2 instância da Amazon na mesma Amazon Virtual Private Cloud (Amazon VPC), o que será o caso deste exercício.

Por padrão, o acesso de rede ao seu cluster é limitado à conta que foi usada para criá-lo. Antes de se conectar a um cluster a partir de uma EC2 instância, você deve autorizar a EC2 instância a acessar o cluster. As etapas necessárias dependem de você ter iniciado seu cluster em EC2 -VPC ou EC2 -Classic.

O caso de uso mais comum é quando um aplicativo implantado em uma EC2 instância precisa se conectar a um cluster na mesma VPC. A maneira mais simples de gerenciar o acesso entre EC2 instâncias e clusters na mesma VPC é fazer o seguinte:

1. Crie um grupo de segurança de VPC para o seu cluster. Esse security group pode ser usado para restringir o acesso às instâncias do cluster. Por exemplo, é possível criar uma regra personalizada para esse grupo de segurança que permite o acesso TCP usando a porta atribuída ao cluster quando você o criou e um endereço IP que será usado para acessar o cluster.

A porta padrão dos clusters e grupos de replicação Redis OSS é 6379.

Important

Os grupos ElastiCache de segurança da Amazon são aplicáveis somente a clusters que não estão sendo executados em um ambiente Amazon Virtual Private Cloud (VPC). Se você estiver executando em uma Amazon Virtual Private Cloud, os Grupos de segurança não estarão disponíveis no painel de navegação do console.

Se você estiver executando seus ElastiCache nós em uma Amazon VPC, você controla o acesso aos seus clusters com grupos de segurança da Amazon VPC, que são diferentes dos grupos de segurança. ElastiCache Para obter mais informações sobre o uso ElastiCache em uma Amazon VPC, consulte [Amazon VPCs e ElastiCache segurança](#)

2. Crie um grupo de segurança VPC para suas EC2 instâncias (servidores web e de aplicativos). Esse grupo de segurança pode, se necessário, permitir o acesso à EC2 instância pela Internet por meio da tabela de roteamento da VPC. Por exemplo, você pode definir regras nesse grupo de segurança para permitir acesso TCP à EC2 instância pela porta 22.
3. Crie regras personalizadas no grupo de segurança do seu cluster que permitam conexões do grupo de segurança que você criou para suas EC2 instâncias. Isso permitiria que qualquer membro de grupo de segurança acessasse os clusters.

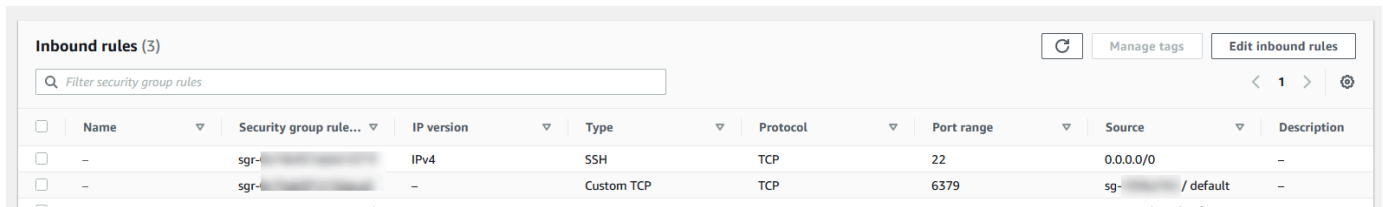
 Note

Se você estiver planejando usar [Uso de zonas locais com o ElastiCache](#), verifique se você as habilitou. Quando você criar um grupo de sub-redes nessa região local, sua VPC será estendida para essa zona local e sua VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade. Todos os gateways e tabelas de rotas relevantes serão ajustados automaticamente.

Para criar uma regra em um grupo de segurança de VPC que permita conexões de outro grupo de segurança

1. [Faça login no AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.](https://console.aws.amazon.com/vpc)
2. No painel de navegação, escolha Security Groups (Grupos de segurança).
3. Selecione ou crie um security group que você usará para suas instâncias de cluster. Em Regras de entrada, selecione Editar regras de entrada e escolha Adicionar regra. Esse grupo de segurança permitirá o acesso a membros de outro grupo de segurança.
4. Em Tipo, escolha Regra TCP personalizada.
 - a. Para Port Range, especifique a porta que você usou quando criou seu cluster.

A porta padrão dos clusters e grupos de replicação Redis OSS é 6379.
 - b. Na caixa Source, comece a digitar o ID do grupo de segurança. Na lista, selecione o grupo de segurança que você usará para suas EC2 instâncias da Amazon.
5. Escolha Save quando terminar.



	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
☐	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
☐	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-

Depois de habilitar o acesso, você agora estará pronto para se conectar ao nó, conforme discutido na próxima seção.

Para obter informações sobre como acessar seu ElastiCache cluster de uma Amazon VPC diferente, de uma AWS região diferente ou até mesmo de sua rede corporativa, consulte o seguinte:

- [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#)
- [Acessando ElastiCache recursos de fora AWS](#)

Etapa 4: conexão ao nó do cluster

Antes de continuar, conclua [Etapa 3: autorizar o acesso ao cluster](#).

Esta seção pressupõe que você criou uma EC2 instância da Amazon e pode se conectar a ela. Para obter instruções sobre como fazer isso, consulte o [Amazon EC2 Getting Started Guide](#).

Uma EC2 instância da Amazon pode se conectar a um nó de cluster somente se você a tiver autorizado a fazer isso.

Localize seus endpoints de nó

Quando seu cluster está no estado disponível e você autorizou o acesso a ele, você pode fazer login em uma EC2 instância da Amazon e se conectar ao cluster. Para isso, primeiro você deve determinar o endpoint.

Localização de endpoints de um cluster do Valkey ou do Redis OSS (modo cluster desabilitado) (console)

Se o cluster do Redis OSS (modo cluster desabilitado) tiver apenas um nó, o endpoint desse nó será usado para tanto para leituras quanto para gravações. Se cluster tiver vários nós, haverá três tipos de endpoints: o endpoint primário, o endpoint leitor e os endpoints de nó.

O endpoint primário é um nome DNS que sempre é resolvido para o nó primário no cluster. O endpoint primário é imune às alterações no seu cluster, como promover uma réplica de leitura para a função primária. Para atividades de gravação, recomendamos que suas aplicações se conectem ao endpoint primário.

Um endpoint de leitura dividirá uniformemente as conexões de entrada com o endpoint entre todas as réplicas de leitura em um cluster OSS ElastiCache for Redis. Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar as várias réplicas de leitura do cluster ElastiCache for Redis OSS em diferentes zonas de AWS disponibilidade (AZ) para garantir a alta disponibilidade dos endpoints de leitura.

Note

Um endpoint leitor não é um balanceador de carga. É um registro DNS que vai resolver para um endereço IP de um dos nós de réplica em uma forma round robin.

Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó no seu aplicativo.

Para localizar um endpoint de um cluster do Redis OSS (modo cluster desabilitado) (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Caches do Redis OSS.

A tela de clusters aparecerá com uma lista que vai incluir todos os caches com tecnologia sem servidor do Valkey ou Redis OSS, clusters do Redis OSS (modo de cluster desabilitado) e do Redis OSS (modo de cluster habilitado). Selecione o cluster que você criou na seção [Criação de um cluster do Redis OSS \(modo cluster desabilitado\) \(console\)](#).

3. Para encontrar os endpoints do Primary and/or Reader do cluster, escolha o nome do cluster (não o botão de opção).

▼ Cluster details			
Cluster name [redacted]	Description [redacted]	Node type cache.r6g.large	Status Available
Engine Redis OSS	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Endpoints primários e leitores para um cluster do Redis OSS (modo cluster desabilitado)

Se houver apenas um nó no cluster, não haverá um endpoint primário, e você poderá continuar na próxima etapa.

4. Se o cluster do Redis OSS (modo cluster desabilitado) tiver nós de réplica, você poderá encontrar seus endpoints dos nós de réplica escolhendo o nome do cluster e escolhendo a guia Nós.

A tela de nós aparece com cada nó do cluster, primário e réplicas, listados com seu respectivo endpoint.

☐	Node Name	Status	Current Role	Port	Endpoint
☐	test-no-001	available	primary	6379	test-no-001.usw2.cache.amazonaws.com:6379
☐	test-no-002	available	replica	6379	test-no-002.usw2.cache.amazonaws.com:6379
☐	test-no-003	available	replica	6379	test-no-003.usw2.cache.amazonaws.com:6379

Endpoint de nó para um cluster do Redis OSS (modo cluster desabilitado)

5. Como copiar um endpoint para a área de transferência:

- Um endpoint por vez, localize o endpoint que você deseja copiar.
- Escolha o ícone de cópia diretamente na frente do endpoint.

O endpoint agora é copiado para a área de transferência. Para obter informações sobre como usar o endpoint para se conectar a um nó, consulte [Conexão a nós](#).

Um endpoint primário do Redis OSS (modo cluster desabilitado) se assemelha ao seguinte. Há uma diferença, dependendo de a criptografia em trânsito estar ou não habilitada.

Criptografia em trânsito não habilitada

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

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

Criptografia em trânsito habilitada

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

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

Para explorar mais sobre como localizar os endpoints, consulte os tópicos relevantes para o tipo de mecanismo e de cluster que você está executando.

- [Encontrar endpoints de conexão no ElastiCache](#)

- [Localização de endpoints para um cluster do Valkey ou do Redis OSS \(modo cluster habilitado\) \(console\)](#): você precisa do endpoint de configuração do cluster.
- [Localizar endpoints \(AWS CLI\)](#)
- [Localização de endpoints \(API do ElastiCache\)](#)

Conexão a um cluster ou grupo de replicação do Valkey ou Redis OSS (Linux)

Agora que você tem o endpoint de que precisa, pode fazer login em uma EC2 instância e se conectar ao cluster ou ao grupo de replicação. No exemplo a seguir, você usa o utilitário valkey-cli para se conectar a um cluster. A versão mais recente do valkey-cli também suporta SSL/TLS a conexão de clusters habilitados. encryption/authentication

O exemplo a seguir usa EC2 instâncias da Amazon executando o Amazon Linux e o Amazon Linux 2. Para obter detalhes sobre como instalar e compilar a valkey-cli com outras distribuições Linux, consulte a documentação do seu sistema operacional específico.

 Note

Este processo abrange o teste de uma conexão usando o utilitário valkey-cli somente para uso não planejado. Para obter uma lista dos clientes compatíveis, consulte a [Documentação do Valkey](#). Para obter exemplos de uso do AWS SDKs com ElastiCache, consulte [Tutoriais: conceitos básicos do Python e do ElastiCache](#).

Conexão a um cluster não criptografado com modo cluster desabilitado

1. Execute o comando a seguir para se conectar ao cluster *primary-endpoint* e substituí-lo *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Valkey e Redis OSS é 6379.)

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

O resultado em um prompt de comando é semelhante ao seguinte:

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

2. Agora você pode executar comandos do Valkey e Redis OSS.

```
set x Hello
OK

get x
"Hello"
```

Conectar a um cluster não criptografado com modo cluster habilitado

1. Execute o comando a seguir para se conectar ao cluster *configuration-endpoint* e substituí-lo *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Valkey e Redis OSS é 6379.)

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

 Note

No comando anterior, a opção -c habilita o modo cluster em seguida aos [redirecionamentos -ASK e -MOVED](#).

O resultado em um prompt de comando é semelhante ao seguinte:

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

2. Agora você pode executar comandos do Valkey e Redis OSS. Observe que o redirecionamento ocorre porque você o habilitou usando a opção -c. Se o redirecionamento não estiver habilitado, o comando retornará o erro MOVED. Para obter mais informações sobre o erro MOVED, consulte [Especificação do cluster](#).

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

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

Conectando-se a um cluster Encryption/Authentication habilitado

Por padrão, a valkey-cli usa uma conexão TCP não criptografada ao se conectar ao Valkey e Redis OSS. A opção é BUILD_TLS=yes SSL/TLS ativada no momento da compilação do valkey-cli, conforme mostrado na seção anterior. [Baixar e configurar o acesso à linha de comando](#) A ativação do AUTH é opcional. No entanto, você deve habilitar a criptografia em trânsito para habilitar o AUTH. Para obter mais detalhes sobre ElastiCache criptografia e autenticação, consulte [ElastiCache criptografia em trânsito \(TLS\)](#).

Note

Você pode usar a opção `--tls` com a valkey-cli para se conectar a clusters criptografados com modo cluster habilitado e desabilitado. Se um cluster tiver um conjunto de tokens AUTH, você poderá usar a opção `-a` para fornecer uma senha do AUTH.

Nos exemplos a seguir, certifique-se de substituir *cluster-endpoint* e *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Redis OSS é 6379.)

Conexão a clusters criptografados com modo cluster desabilitado

O exemplo a seguir conecta a um cluster habilitado para criptografia e autenticação:

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

O exemplo a seguir conecta a um cluster somente habilitado para criptografia:

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

Conexão a clusters criptografados com modo cluster habilitado

O exemplo a seguir conecta a um cluster habilitado para criptografia e autenticação:

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

O exemplo a seguir conecta a um cluster somente habilitado para criptografia:

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

Depois de se conectar ao cluster, você poderá executar os comandos do Valkey ou Redis OSS, conforme mostrado nos exemplos anteriores para clusters não criptografados.

Uma alternativa ao valkey-cli ou ao Redis-cli

Se o cluster não estiver habilitado para o modo cluster e você precisar fazer uma conexão com o cluster para um teste curto, mas sem passar pela compilação da valkey-cli ou redis-cli, você pode usar telnet ou openssl. Nos comandos de exemplo a seguir, certifique-se de substituir *cluster-endpoint* e *port number* com o endpoint do seu cluster e o número da sua porta. (A porta padrão para o Redis OSS é 6379.)

O exemplo a seguir se conecta a um cluster desativado no modo de cluster habilitado para and/or autenticação por criptografia:

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

Se o cluster tiver uma senha definida, conecte-se primeiro ao cluster. Após a conexão, autentique o cluster usando o seguinte comando, depois pressione a tecla Enter. No exemplo a seguir, *your-password* substitua pela senha do seu cluster.

```
Auth your-password
```

O exemplo a seguir conecta a um cluster com modo cluster desabilitado que não tenha criptografia ou autenticação habilitada:

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

Conexão a um cluster ou grupo de replicação do Valkey ou Redis OSS (Windows)

Para se conectar ao cluster a partir de uma instância do EC2 Windows usando a CLI do OSS Valkey ou Redis, você deve baixar o pacote valkey-cli e usar o valkey-cli.exe para se conectar ao cluster OSS Valkey ou Redis a partir de uma instância do Windows. EC2

No exemplo a seguir, use o utilitário `valkey-cli` para se conectar a um cluster que não está habilitado para criptografia e que está executando o Valkey ou Redis OSS. Para obter mais informações sobre o Valkey e os comandos disponíveis, consulte [Comandos do Valkey](#) no site do Valkey.

Para se conectar a um cluster do Valkey ou Redis OSS não habilitado para criptografia usando o `valkey-cli`

1. Conecte-se à sua EC2 instância da Amazon usando o utilitário de conexão de sua escolha. Para obter instruções sobre como se conectar a uma EC2 instância da Amazon, consulte o [Amazon EC2 Getting Started Guide](#).
2. Copie e cole o link <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> em um navegador da Internet para baixar o arquivo zip para o cliente Redis OSS a partir da versão disponível em GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extraia o arquivo zip para a pasta/caminho desejado.

Abra o prompt de comando, mude para o diretório do Valkey e execute o comando `c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379`.

Por exemplo:

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

3. Execute os comandos Valkey ou Redis OSS.

Agora, você está conectado ao cluster e pode executar comandos Valkey ou Redis OSS como os seguintes.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5   // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                   // Get value for key "b"
"Good-bye"
                        // wait >= 5 seconds
get b
```

```
(nil)          // key has expired, nothing returned
quit          // Exit from valkey-cli
```

Excluir um cluster

Enquanto um cluster estiver no estado disponível, você será cobrado por ele, independentemente de o estar ou não. Para interromper as cobranças, exclua o cluster.

Warning

- Quando você exclui um ElastiCache cluster, seus instantâneos manuais são retidos. Também é possível criar um snapshot final antes que o cluster seja excluído. Os snapshots de cache automáticos não são retidos. Para obter mais informações, consulte [Snapshots e restauração](#).
- É necessário ter a permissão `CreateSnapshot` para criar um snapshot final. Sem essa permissão, a chamada de API falhará com uma exceção `Access Denied`.

Usando o Console de gerenciamento da AWS

O procedimento a seguir exclui um único cluster da sua implantação. Para excluir vários clusters, repita o procedimento para cada cluster que deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster antes de iniciar o procedimento para excluir outro.

Para excluir um cluster

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache motor, escolha Valkey ou Redis OSS.

Uma lista de todos os caches em execução naquele mecanismo é exibida.

3. Para escolher o cluster a ser excluído, escolha o nome na lista de clusters. Nesse caso, o nome do cluster do que você criou em [Etapa 2: criar um cluster](#).

 Important

Você só pode excluir um cluster por vez do ElastiCache console. Escolher vários clusters desabilita a operação de exclusão.

4. Em Ações, escolha Excluir.
5. Na tela de confirmação Excluir cluster, digite o nome do cluster e escolha Backup final. Em seguida, escolha Excluir para excluir o cluster ou escolha Cancelar para manter o cluster.

Se você escolheu Excluir, o status do cluster muda para excluindo.

Assim que o cluster não estiver mais relacionado na lista de clusters, você para de ser cobrado por ele.

Usando o AWS CLI

O código a seguir exclui o cluster `my-cluster`. Neste caso, substitua `my-cluster` pelo nome do cluster do que você criou em [Etapa 2: criar um cluster](#).

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

A ação `delete-cache-cluster` da CLI apenas exclui um cluster. Para excluir vários clusters, chame `delete-cache-cluster` para cada cluster de cache que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster antes de excluir outro.

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

Para Windows:

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

Para obter mais informações, consulte o ElastiCache tópico AWS CLI for [delete-cache-cluster](#).

Outros tutoriais e vídeos do ElastiCache

Os seguintes tutoriais discutem tarefas de interesse para o usuário do Amazon ElastiCache.

- [Vídeos do ElastiCache](#)
- [Tutorial: configurar uma função Lambda para acessar o Amazon ElastiCache em uma Amazon VPC](#)

Vídeos do ElastiCache

Aqui estão vídeos para ajudar você a aprender os conceitos básicos e avançados do Amazon ElastiCache. Para obter mais informações sobre treinamento da AWS, consulte [Treinamento e certificação da AWS](#).

Tópicos

- [Vídeos de introdução](#)
- [Vídeos avançados](#)

Vídeos de introdução

Os seguintes vídeos apresentam você ao Amazon ElastiCache.

Tópicos

- [AWS re:Invent 2020: o que há de novo no Amazon ElastiCache](#)
- [AWS re:Invent 2019: o que há de novo no Amazon ElastiCache](#)
- [AWS re:Invent 2017: o que há de novo no Amazon ElastiCache](#)
- [DAT204: Criação de aplicações escaláveis em serviços NoSQL da AWS \(re:Invent 2015\)](#)
- [DAT207: Aceleração de performance da aplicação com o Amazon ElastiCache \(AWS re:Invent 2013\)](#)

AWS re:Invent 2020: o que há de novo no Amazon ElastiCache

[AWS re:Invent 2020: o que há de novo no Amazon ElastiCache](#)

AWS re:Invent 2019: o que há de novo no Amazon ElastiCache

[AWS re:Invent 2019: o que há de novo no Amazon ElastiCache](#)

AWS re:Invent 2017: o que há de novo no Amazon ElastiCache

[AWS re:Invent 2017: o que há de novo no Amazon ElastiCache](#)

DAT204: Criação de aplicações escaláveis em serviços NoSQL da AWS (re:Invent 2015)

Nesta sessão, discutimos os benefícios dos bancos de dados NoSQL e fazemos um tour pelos principais serviços NoSQL oferecidos pela AWS: Amazon DynamoDB e Amazon ElastiCache. Em

seguida, ouvimos dois clientes líderes, a Expedia e a Mapbox, sobre seus casos de uso e desafios arquitetônicos e como eles os enfrentaram usando os serviços NoSQL da AWS, incluindo padrões de design e melhores práticas. Você sairá dessa sessão com uma melhor compreensão sobre o NoSQL e seus poderosos recursos, pronto para enfrentar seus desafios de banco de dados com confiança.

[DAT204: Criação de aplicações escaláveis em serviços NoSQL da AWS \(re:Invent 2015\)](#)

DAT207: Aceleração de performance da aplicação com o Amazon ElastiCache (AWS re:Invent 2013)

Neste vídeo, saiba como você pode usar o Amazon ElastiCache para implantar facilmente um sistema de cache de memória para acelerar a performance da aplicação. Mostramos como usar o Amazon ElastiCache para melhorar a latência da aplicação e reduzir a carga nos seus servidores de banco de dados. Também mostramos como criar uma camada de cache que é fácil de gerenciar e escalar à medida que o seu aplicativo se expande. Durante esta sessão, examinamos vários cenários e casos de uso que podem se beneficiar com a habilitação do armazenamento em cache, além de discutirmos os recursos fornecidos pelo Amazon ElastiCache.

[DAT207: Aceleração de performance de aplicação com o Amazon ElastiCache \(re:Invent 2013\)](#)

Vídeos avançados

Os seguintes vídeos abrangem tópicos mais avançados do Amazon ElastiCache.

Tópicos

- [Design para o sucesso com as práticas recomendadas do Amazon ElastiCache \(re:Invent 2020\)](#)
- [Sobrecarregue suas aplicações em tempo real com o Amazon ElastiCache \(re:Invent 2019\)](#)
- [Práticas recomendadas: migração de clusters Redis OSS do Amazon EC2 para o ElastiCache \(re:Invent 2019\)](#)
- [Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Redis OSS confiável e escalável na nuvem com o Amazon ElastiCache \(re:Invent 2018\)](#)
- [Aprofundamento no ElastiCache: padrões de design para armazenamentos de dados na memória \(re:Invent 2018\)](#)
- [DAT305: Aprofundamento no Amazon ElastiCache \(re:Invent 2017\)](#)
- [DAT306: Aprofundamento no Amazon ElastiCache \(re:Invent 2016\)](#)
- [DAT317: como o IFTTT usa o ElastiCache para Redis OSS a fim de prever eventos \(re:Invent 2016\)](#)

- [DAT407: Aprofundamento no Amazon ElastiCache \(re:Invent 2015\)](#)
- [SDD402: Aprofundamento no Amazon ElastiCache \(re:Invent 2014\)](#)
- [DAT307: Aprofundamento na arquitetura e nos padrões de design do Amazon ElastiCache \(re:Invent 2013\).](#)

Design para o sucesso com as práticas recomendadas do Amazon ElastiCache (re:Invent 2020)

Com o crescimento explosivo de aplicações críticas para os negócios e em tempo real criadas com base no Redis OSS, a disponibilidade, a escalabilidade e a segurança se tornaram as principais considerações. Conheça as melhores práticas para configurar o Amazon ElastiCache para obter sucesso com escalabilidade on-line, alta disponibilidade em implantações multi-AZ e configurações de segurança.

[Design para o sucesso com as práticas recomendadas do Amazon ElastiCache \(re:Invent 2020\)](#)

Sobrecarregue suas aplicações em tempo real com o Amazon ElastiCache (re:Invent 2019)

Com o rápido crescimento da adoção da nuvem e os novos cenários que ela capacita, as aplicações precisam de latência de microssegundos e alto throughput para suportar milhões de solicitações por segundo. Tradicionalmente, os desenvolvedores contam com hardware especializado e soluções alternativas, como bancos de dados baseados em disco combinados com técnicas de redução de dados, para gerenciar dados para aplicações em tempo real. Essas abordagens podem ser caras e não escaláveis. Saiba como aumentar a performance de aplicações em tempo real usando o Amazon ElastiCache totalmente gerenciado e na memória para obter performance extrema, alta escalabilidade, disponibilidade e segurança.

[Sobrecarregue suas aplicações em tempo real com o Amazon ElastiCache \(re:Invent 2019:\)](#)

Práticas recomendadas: migração de clusters Redis OSS do Amazon EC2 para o ElastiCache (re:Invent 2019)

Gerenciar clusters Redis OSS por conta própria pode ser difícil. Você precisa provisionar hardware, corrigir software, fazer backup de dados e monitorar cargas de trabalho constantemente. Com o recém-lançado recurso de migração online para o Amazon ElastiCache, agora você pode mover facilmente seus dados do Redis OSS auto-hospedado no Amazon EC2 para o Amazon ElastiCache totalmente gerenciado, com o modo cluster desabilitado. Nesta sessão, você aprenderá sobre a nova ferramenta de migração online, verá uma demonstração e, mais importante, aprenderá práticas recomendadas para uma migração tranquila para o Amazon ElastiCache.

[Práticas recomendadas: migração de clusters Redis OSS do Amazon EC2 para o ElastiCache \(re:Invent 2019\)](#)

Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora STP11 (re:Invent 2018)

Dream11 é a principal startup de tecnologia esportiva da Índia. Ela tem uma base crescente de mais de 40 milhões de usuários jogando vários esportes, incluindo críquete, futebol e basquete de fantasia, e atualmente atende a um milhão de usuários simultâneos, que produzem três milhões de solicitações por minuto sob um tempo de resposta de 50 milissegundos. Nesta palestra, o CTO da Dream11, Amit Sharma, explica como a empresa usa o Amazon Aurora e o Amazon ElastiCache para lidar com o tráfego instantâneo, que pode triplicar em uma janela de resposta de 30 segundos. Sharma também fala sobre escalar transações sem bloquear, e ele compartilha as etapas para lidar com o tráfego instantâneo, servindo assim a cinco milhões de usuários ativos diários. Título completo: AWS re:Invent 2018: Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora (STP11)

[Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora \(STP11\) \(re:Invent 2018\)](#)

Redis OSS confiável e escalável na nuvem com o Amazon ElastiCache (re:Invent 2018)

Esta sessão aborda os atributos e aprimoramentos do nosso serviço compatível com Redis OSS, Amazon ElastiCache para Redis OSS. Cobrimos os principais recursos, como Redis OSS 5, melhorias de escalabilidade e performance, segurança e conformidade e muito mais. Também discutimos recursos futuros e estudos de caso de clientes.

[Redis OSS confiável e escalável na nuvem com o Amazon ElastiCache \(re:Invent 2018\)](#)

Aprofundamento no ElastiCache: padrões de design para armazenamentos de dados na memória (re:Invent 2018)

Nesta sessão, fornecemos uma visão dos bastidores para saber mais sobre o design e a arquitetura do Amazon ElastiCache. Veja os padrões de design comuns com nossas ofertas do Redis OSS e do Memcached e como os clientes os usam para processamento de dados na memória para reduzir a latência e melhorar o throughput da aplicação. Revisamos as melhores práticas, os padrões de design e os antipadrões do ElastiCache.

[Aprofundamento no ElastiCache: padrões de design para armazenamentos de dados na memória \(re:Invent 2018\)](#)

DAT305: Aprofundamento no Amazon ElastiCache (re:Invent 2017)

Confira os bastidores para saber mais sobre o design e a arquitetura da Amazon ElastiCache. Veja os padrões de design comuns com nossas ofertas do Memcached e do Redis OSS e como os clientes os usaram para operações na memória para reduzir a latência e melhorar o throughput do aplicativo. Neste vídeo, revisamos as melhores práticas, os padrões de design e os antipadrões do ElastiCache.

O vídeo apresenta o seguinte:

- Refragmentação online do ElastiCache para Redis OSS
- Segurança e criptografia do ElastiCache
- ElastiCache para Redis OSS versão 3.2.10

[DAT305: Aprofundamento no Amazon ElastiCache \(re:Invent 2017\)](#)

DAT306: Aprofundamento no Amazon ElastiCache (re:Invent 2016)

Confira os bastidores para saber mais sobre o design e a arquitetura da Amazon ElastiCache. Veja os padrões de design comuns com nossas ofertas do Memcached e do Redis OSS e como os clientes os usaram para operações na memória para reduzir a latência e melhorar o throughput do aplicativo. Durante esta sessão, analisamos as melhores práticas, os padrões de design e os antipadrões relacionados ao ElastiCache.

[DAT306: Aprofundamento no Amazon ElastiCache \(re:Invent 2016\)](#)

DAT317: como o IFTTT usa o ElastiCache para Redis OSS a fim de prever eventos (re:Invent 2016)

O IFTTT é um serviço gratuito que permite às pessoas fazer mais com seus serviços preferidos, desde automatizar tarefas simples até transformar a forma como alguém interage e controla sua casa. O IFTTT usa o ElastiCache para Redis OSS a fim de armazenar o histórico de execução de transações e as previsões de programação, bem como índices de documentos de log no Amazon S3. Assista a esta sessão para saber como o poder de scripts do Lua e os tipos de dados do Redis OSS permitiram que as pessoas realizassem algo que não poderiam ter feito em nenhum outro lugar.

[DAT317: como o IFTTT usa o ElastiCache para Redis OSS a fim de prever eventos \(re:Invent 2016\)](#)

DAT407: Aprofundamento no Amazon ElastiCache (re:Invent 2015)

Olhe nos bastidores para saber mais sobre o design e a arquitetura da Amazon ElastiCache. Veja os padrões de design comuns das nossas ofertas Memcached e Redis OSS e como os clientes

os usaram para operações na memória e obtiveram níveis aprimorados de latência e throughput para aplicativos. Durante esta sessão, analisamos as melhores práticas, os padrões de design e os antipadrões relacionados ao Amazon ElastiCache.

[DAT407: Aprofundamento no Amazon ElastiCache \(re:Invent 2015\)](#)

[SDD402: Aprofundamento no Amazon ElastiCache \(re:Invent 2014\)](#)

Neste vídeo, examinamos os casos de uso comuns de armazenamento em cache, os mecanismos ElastiCache para Memcached e ElastiCache para Redis OSS, os padrões que o ajudam a determinar qual mecanismo é o melhor para suas necessidades, aplicação de hash consistente e muito mais como de compilar aplicativos rápidos e escaláveis. Frank Wiebe, o principal cientista da Adobe, detalha como a Adobe usa o Amazon ElastiCache para melhorar a experiência dos clientes e escalonar seus negócios.

[DAT402: Aprofundamento no Amazon ElastiCache \(re:Invent 2014\)](#)

DAT307: Aprofundamento na arquitetura e nos padrões de design do Amazon ElastiCache (re:Invent 2013).

Neste vídeo, examinamos o armazenamento em cache, as estratégias de cache, a expansão e o monitoramento. Comparamos também os mecanismos ElastiCache para Memcached e ElastiCache para Redis OSS. Durante esta sessão, também revisamos as melhores práticas e os padrões de design relacionados ao Amazon ElastiCache.

[DAT307: Aprofundamento na arquitetura e nos padrões de design do Amazon ElastiCache \(AWS re:Invent 2013\).](#)

Gerenciamento de nós no ElastiCache

Um nó é o menor bloco de construção de uma implantação do Amazon ElastiCache. Trata-se de um fragmento de tamanho fixo da RAM em armazenamento de rede anexado. Cada nó executa o mecanismo que foi escolhido quando o cluster foi criado ou modificado pela última vez. Cada nó possui seu próprio nome DNS (Serviço de Nomes de Domínio) e porta. Há suporte para vários tipos de nós do ElastiCache, cada um com quantidades diversificadas de memória associada e a potência computacional.

Consulte para uma discussão mais detalhada sobre o tamanho do nó que deve ser usado [Escolha do tamanho do nó](#).

Em termos gerais, devido ao seu suporte para fragmentação, as implantações do Valkey ou Redis OSS (modo cluster habilitado) têm vários nós menores. Por outro lado, as implantações do Valkey ou Redis OSS (modo cluster desabilitado) têm menos nós, porém maiores, em um cluster. Consulte para uma discussão mais detalhada sobre o tamanho do nó que deve ser usado [Escolha do tamanho do nó](#).

Tópicos

- [Visualizando o status ElastiCache do nó](#)
- [Nódulos e fragmentos do Valkey ou Redis OSS](#)
- [Conexão a nós](#)
- [Tipos de nó compatíveis](#)
- [Reinicialização de nós](#)
- [Substituição de nós \(Valkey e Redis OSS\)](#)
- [Substituição de nós \(Memcached\)](#)
- [Nós reservados](#)
- [Migração de nós da geração anterior](#)

Algumas operações importantes que envolvem nós são as seguintes:

- [Adicionar nós a um ElastiCache cluster](#)
- [Removendo nós de um ElastiCache cluster](#)
- [Dimensionamento ElastiCache](#)
- [Encontrar endpoints de conexão no ElastiCache](#)
- [Identificar automaticamente nós no seu cluster \(Memcached\)](#)

Visualizando o status ElastiCache do nó

Usando o [ElastiCache console](#), você pode acessar rapidamente o status do seu ElastiCache nó. O status de um ElastiCache nó indica a integridade do nó. Você pode usar os procedimentos a seguir para visualizar o status do ElastiCache nó no ElastiCache console da Amazon, no AWS CLI comando ou na operação da API.

Os valores de status possíveis para ElastiCache os nós estão na tabela a seguir. Essa tabela também mostra se você será cobrado pelo ElastiCache nó.

Tipo	Faturado	Descrição
available	Faturado	O ElastiCache nóduo está saudável e disponível.
creating	Não faturado	O ElastiCache nó está sendo criado. O nó está inacessível enquanto é criado.
deleting	Não faturado	O ElastiCache nó está sendo excluído.
modifying	Faturado	O ElastiCache nó está sendo modificado devido a uma solicitação do cliente para modificar o nó.
updating	Faturado	Um estado de atualização indica que uma ou mais das seguintes afirmações são verdadeiras para o ElastiCache nó da Amazon: • O ElastiCache nó está sendo corrigido como parte da atualização do serviço. Para obter mais informações sobre as atualizações do serviço, consulte a página de ajuda do Amazon ElastiCache Managed Maintenance and Service Updates. • Os grupos de segurança da VPC estão sendo atualizados para o ElastiCache cluster.

Tipo	Faturado	Descrição
		• O ElastiCache cluster está sendo ampliado ou reduzido. • As configurações de entrega de registros estão sendo modificadas para o ElastiCache cluster. • Uma operação de exclusão do ElastiCache nó está pendente. • A ElastiCache senha está sendo updated/rotated usada AWS Secrets Manager.
rebooting cluster nodes	Faturado	O ElastiCache nó está sendo reinicializado devido a uma solicitação do cliente ou a um ElastiCache processo da Amazon que exige a reinicialização do nó.

Tipo	Faturado	Descrição
<code>incompatible_parameters</code>	Não faturado	A Amazon não ElastiCache pode iniciar o nó porque os parâmetros especificados no grupo de parâmetros do nó não são compatíveis com o nó. Reverta as alterações de parâmetro ou as torne compatíveis com o nó para retomar o acesso ao nó. Para obter mais informações sobre os parâmetros incompatíveis, consulte a lista de eventos do ElastiCache nó.
<code>incompatible_network</code>	Não faturado	Um estado de rede incompatível indica que uma ou mais das seguintes afirmações são verdadeiras para o nó da Amazon: ElastiCache • Não há endereços IP disponíveis na sub-rede em que o ElastiCache nó foi lançado. • A sub-rede mencionada no grupo de ElastiCache sub-redes não existe mais na Amazon Virtual Private Cloud (Amazon VPC).

Tipo	Faturado	Descrição
restore_failed	Não faturado	Um estado de falha na restauração indica que uma das seguintes situações se aplica ao nó da Amazon ElastiCache • As substituições de nó com falha por causa da capacidade da instância insuficiente. Isso normalmente acontece ao executar nós da geração anterior que são end-of-life. No entanto, isso também pode acontecer com a substituição dos nós da geração atual quando AWS não há capacidade sob demanda suficiente para atender à sua solicitação na zona de disponibilidade especificada. Para obter mais informações sobre como corrigir ou remover esses nós, consulte Migração de nós da geração anterior. • Falha na restauração do instantâneo do RDB especificado. • A AWS conta do ElastiCache cluster foi suspensa. • O nó falhou e não pôde ser recuperado.

Tipo	Faturado	Descrição
snapshotting	Faturado	ElastiCache está criando um instantâneo do nó OSS Valkey ou Redis.

Visualizando o status do ElastiCache nó com o console

Para ver o status de um ElastiCache Node com o console:

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Redis OSS ou Clusters do Memcached. A página Caches aparece com a lista de ElastiCache nós. Para cada nó, o valor do status é exibido.
3. Em seguida, você pode navegar até a guia Atualizações de serviço do cache para exibir a lista de atualizações de serviço aplicáveis ao cache.

Visualizando o status do ElastiCache nó com o AWS CLI

Para visualizar o ElastiCache nó e suas informações de status usando o AWS CLI, use o `describe-cache-cluster` comando. Por exemplo, o AWS CLI comando a seguir exibe cada ElastiCache nó.

```
aws elasticache describe-cache-clusters
```

Visualizando o status do ElastiCache nó por meio da API

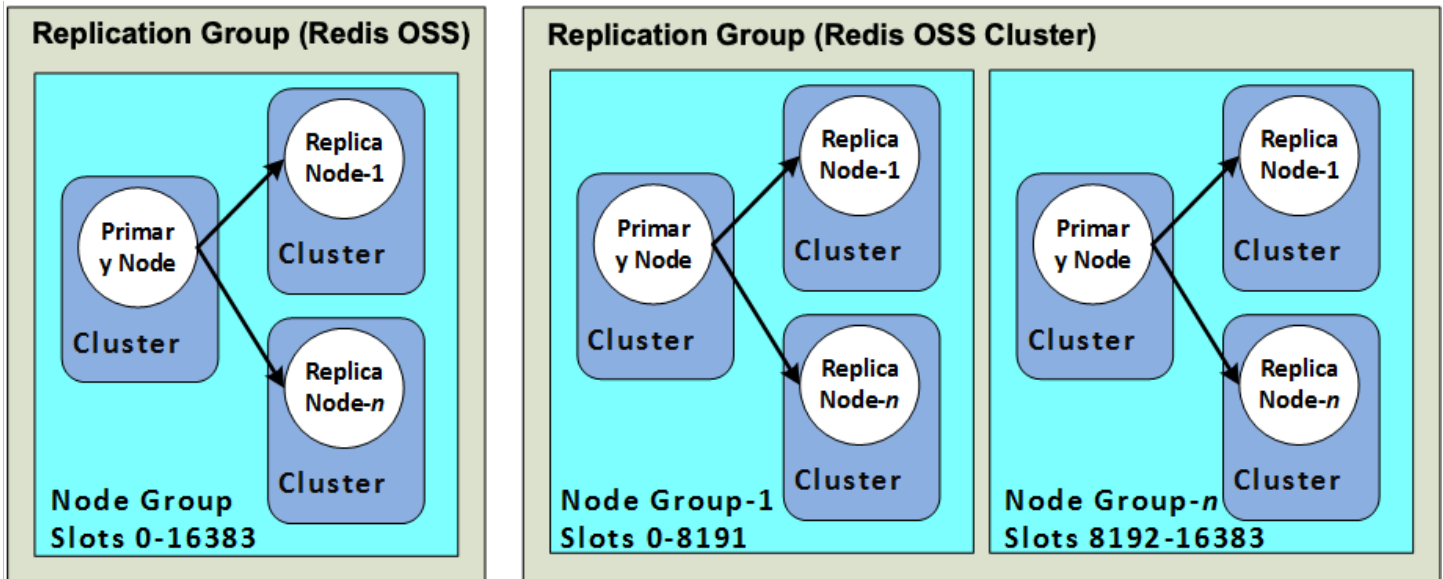
Para visualizar o status do ElastiCache nó usando a ElastiCache API da Amazon, chame `DescribeCacheClusteroperation` with the `ShowCacheNodeInfo` flag para recuperar informações sobre os nós de cache individuais.

Nódulos e fragmentos do Valkey ou Redis OSS

Um fragmento (na API e na CLI, um grupo de nós) é uma disposição hierárquica de nós, cada um envolvido em um cluster. Fragmentos oferecem suporte para replicação. Dentro de um fragmento, um nó funciona como o nó primário de leitura/gravação. Todos os outros nós em um fragmento funcionam como réplicas somente leitura do nó primário. O Valkey, ou o Redis OSS versão 3.2 e

posteriores, oferecem suporte a vários fragmentos dentro de um cluster (na API e CLI, um grupo de replicação). Esse suporte permite particionar os dados em um cluster do Valkey ou Redis OSS (modo cluster habilitado).

O diagrama a seguir ilustra as diferenças entre um cluster do Valkey ou Redis OSS (modo cluster desabilitado) e um cluster do Valkey ou Redis OSS (modo cluster habilitado).



Os clusters do Valkey ou Redis OSS (modo cluster habilitado) oferecem suporte para a replicação por meio de fragmentos. A operação de API, [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) lista os grupos de nós com nós membros, a função do nó dentro do grupo de nós, bem como outras informações.

Ao criar um cluster do Valkey ou Redis OSS, você especifica se deseja criar um cluster com o recurso de criação de clusters habilitado. Os clusters do Valkey ou Redis OSS (modo cluster desabilitado) nunca têm mais de um fragmento que pode ser escalado horizontalmente com a adição (até um total de cinco) ou a exclusão de nós de réplica de leitura. Para mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#), [Adição de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#) ou [Exclusão de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#). Clusters do Valkey ou Redis OSS (modo cluster desabilitado) também podem ser escalados verticalmente por meio da mudança dos tipos de nó. Para obter mais informações, consulte [Escalar nós de réplica para Valkey ou Redis OSS \(modo cluster desativado\)](#).

O limite de nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se o mecanismo for Valkey ou Redis OSS versão 5.0.6 ou superior. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP

disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Depois que um cluster do Valkey ou Redis OSS (modo cluster habilitado) é criado, ele pode ser alterado (com aumento ou redução de escala na horizontal). Para obter mais informações, consulte [Dimensionamento ElastiCache](#) e [Substituição de nós \(Valkey e Redis OSS\)](#).

Ao criar um novo cluster, você pode preenchê-lo com dados do cluster antigo para que ele não fique vazio. Essa abordagem só funcionará se o grupo de clusters tiver o mesmo número de estilhaços do cluster antigo. Isso pode ser útil se você precisa alterar o tipo de nó ou a versão do mecanismo. Para obter mais informações, consulte [Realização de backups manuais](#) e [Restauração de um backup para um novo cache](#).

Conexão a nós

Conexão a nós do Valkey ou Redis OSS

Antes de tentar se conectar aos nós do Valkey ou Redis OSS no cluster, você deve ter os endpoints para os nós. Para localizar os endpoints, consulte o seguinte:

- [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para um cluster do Valkey ou do Redis OSS \(modo cluster habilitado\) \(console\)](#)
- [Localizar endpoints \(AWS CLI\)](#)
- [Localização de endpoints \(API do ElastiCache\)](#)

No exemplo a seguir, você usa o utilitário valkey-cli para se conectar a um cluster que está executando o Valkey ou Redis OSS.

Note

Para obter mais informações sobre os comandos disponíveis, consulte a página da Web [Comandos](#).

Para se conectar a um cluster do Valkey ou Redis OSS usando o valkey-cli

1. Conexão à sua instância do Amazon EC2 com o utilitário de conexão de sua escolha.

Note

Para obter instruções sobre como se conectar a uma instância do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon EC2](#).

2. Para criar valkey-cli, baixe e instale o GNU Compiler Collection (gcc). No prompt de comando da sua instância do EC2, digite o seguinte comando e digite y no prompt de confirmação.

```
sudo yum install gcc
```


É exibida uma saída semelhante à seguinte.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB    00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm              | 4.8 MB    00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm                | 2.8 kB    00:00

...(output omitted)...

Complete!
```

3. Baixe e compile o utilitário valkey-cli. Esse utilitário está incluído na distribuição do software Valkey. No prompt de comando da sua instância do EC2, digite os seguintes comandos:

 Note

Para sistemas Ubuntu, antes de executar o make, execute `make distclean`.

```
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make distclean      # ubuntu systems only
make
```

4. No prompt de comando da sua instância do EC2, digite o comando a seguir.

```
src/valkey-cli -c -h mycachecuster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

É exibido um prompt de comando do Valkey ou Redis OSS semelhante ao seguinte.

```
redis mycacheccluster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

5. Teste a conexão executando comandos do Valkey ou Redis OSS.

Agora, você está conectado ao cluster e pode executar comandos do Valkey ou Redis OSS. Os seguintes são alguns comandos de exemplo com suas respostas do Valkey ou Redis OSS.

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

Para conectar-se a nós ou clusters com criptografia Secure Sockets Layer (SSL) (em trânsito habilitado), consulte [ElastiCache criptografia em trânsito \(TLS\)](#).

Conexão a nós do Memcached

Antes de tentar se conectar ao seu cluster Memcached, você deve ter os endpoints para os nós. Para localizar os endpoints, consulte o seguinte:

- [Localização de endpoints de um cluster \(console\) \(Memcached\)](#)
- [Localizar endpoints \(AWS CLI\)](#)
- [Localização de endpoints \(API do ElastiCache\)](#)

No exemplo a seguir, você usa o utilitário telnet para se conectar a um nó que está executando o Memcached.

Note

Para obter mais informações sobre o Memcached e os comandos Memcached disponíveis, consulte o site do [Memcached](#).

Para se conectar a um nó usando o telnet

1. Conecte-se à instância do Amazon EC2 com o utilitário de conexão de sua escolha.

Note

Para obter instruções sobre como se conectar a uma instância do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon EC2](#).

2. Baixe e instale o utilitário telnet na sua instância do Amazon EC2. No prompt de comando da sua instância do Amazon EC2, digite o seguinte comando e digite y.

```
sudo yum install telnet
```

É exibida uma saída semelhante à seguinte.

```
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. No prompt de comando da sua instância do Amazon EC2, digite o seguinte comando, substituindo o endpoint do seu nó por aquele mostrado neste exemplo.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

É exibida uma saída semelhante à seguinte.

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

4. Teste a conexão executando comandos Memcached.

Agora você está conectado a um nó e pode executar comandos Memcached. Veja um exemplo a seguir.

```
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
>
```

Tipos de nó compatíveis

ElastiCache suporta os seguintes tipos de nós. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das gerações anteriores.

Para obter mais informações sobre detalhes de desempenho para cada tipo de nó, consulte [Tipos de EC2 instância da Amazon](#).

Note

A Amazon ElastiCache está fazendo a transição de instâncias T2 para o status de geração anterior. Você não poderá mais criar novos ElastiCache clusters usando instâncias T2 ou comprar novos nós reservados T2. Não há impacto nos clusters ou reservas T2 existentes. Recomendamos a atualização para tipos de instância mais novos, como instâncias T3 ou T4G, para melhorar o desempenho e a economia.

Note

Os seguintes tipos de instância são compatíveis com as AWS regiões Ásia-Pacífico (Tailândia) e México (Central):

- m7g/r7g: grande, xl, 2xl, 4xl, 8xl, 12xl e 16xl.
- t3/t4g: micro, pequeno e médio.

Para obter informações sobre o tamanho do nó que deve ser usado, consulte [Escolha do tamanho do nó](#).

Tópicos

- [Geração atual \(Memcached\)](#)
- [Geração atual \(Valkey ou Redis OSS\)](#)
- [Tipos de nós compatíveis por região da AWS](#)
- [Instâncias de desempenho com capacidade de intermitência](#)
- [Informações relacionadas](#)

Geração atual (Memcached)

As tabelas a seguir mostram a largura de banda básica e de intermitência dos tipos de instância que usam o mecanismo de I/O crédito de rede para ultrapassar sua largura de banda básica.

Note

Os tipos de instância com desempenho de rede intermitente usam um mecanismo I/O de crédito de rede para ultrapassar sua largura de banda básica com base no melhor esforço.

Geral

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m7g.large		0,937	12,5
cache.m7g.xlarge		1.876	12,5
cache.m7g.2xlarge		3,75	15
cache.m7g.4xlarge		7,5	15
cache.m7g.8xlarge		15	N/D
cache.m7g.12xlarge		22,5	N/D
cache.m7g.16xlarge		30	N/D
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/D

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m6g.12xlarge	1.5.16	20	N/D
cache.m6g.16xlarge	1.5.16	25	N/D
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/D	N/D
cache.m5.24xlarge	1.5.16	N/D	N/D
cache.m4.large	1.5.16	0,45	1.2
cache.m4.xlarge	1.5.16	0.75	2.8
cache.m4.2xlarge	1.5.16	1,0	10.0
cache.m4.4xlarge	1.5.16	2,0	10.0
cache.m4.10xlarge	1.5.16	5,0	10.0
cache.t4g.micro	1.5.16	0,064	5,0
cache.t4g.small	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

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
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

Memória otimizada para o Memcached

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r7g.large		0,937	12,5
cache.r7g.xlarge		1.876	12,5
cache.r7g.2xlarge		3,75	15
cache.r7g.4xlarge		7,5	15
cache.r7g.8xlarge		15	N/D
cache.r7g.12xlarge		22,5	N/D
cache.r7g.16xlarge		30	N/D
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/D
cache.r6g.12xlarge	1.5.16	20	N/D

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6g.16xlarge	1.5.16	25	N/D
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/D
cache.r5.24xlarge	1.5.16	25	N/D
cache.r4.large	1.5.16	0.75	10.0
cache.r4.xlarge	1.5.16	1,25	10.0
cache.r4.2xlarge	1.5.16	2,5	10.0
cache.r4.4xlarge	1.5.16	5,0	10.0
cache.r4.8xlarge	1.5.16	12	N/D
cache.r4.16xlarge	1.5.16	25	N/D

Rede otimizada para o Memcached

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
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

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.c7gn.4xlarge	1.6.6	50	N/D
cache.c7gn.8xlarge	1.6.6	100	N/D
cache.c7gn.12xlarge	1.6.6	150	N/D
cache.c7gn.16xlarge	1.6.6	200	N/D

Geração atual (Valkey ou Redis OSS)

Para obter mais informações sobre a geração anterior, consulte [Nós da geração anterior](#).

Note

Os tipos de instância com desempenho de rede intermitente usam um mecanismo I/O de crédito de rede para ultrapassar sua largura de banda básica com base no melhor esforço.

Geral

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Multiplexação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m7g.large	6.2	N	N	N	0,937	12,5
cache.m7g.xlarge	6.2	S	S	S	1.876	12,5

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Multiplexação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m7g.2xlarge	6.2	S	S	S	3,75	15
cache.m7g.4xlarge	6.2	S	S	S	7,5	15
cache.m7g.8xlarge	6.2	S	S	S	15	N/D
cache.m7g.12xlarge	6.2	S	S	S	22,5	N/D
cache.m7g.16xlarge	6.2	S	S	S	30	N/D
cache.m6g.large	5.0.6	N	N	N	0.75	10.0
cache.m6g.xlarge	5.0.6	S	S	S	1,25	10.0
cache.m6g.2xlarge	5.0.6	S	S	S	2,5	10.0
cache.m6g.4xlarge	5.0.6	S	S	S	5,0	10.0
cache.m6g.8xlarge	5.0.6	S	S	S	12	N/D

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Multiplexação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m6g.12xlarge	5.0.6	S	S	S	20	N/D
cache.m6g.16xlarge	5.0.6	S	S	S	25	N/D
cache.m5.large	3.2.4	N	N	N	0.75	10.0
cache.m5.xlarge	3.2.4	S	N	N	1,25	10.0
cache.m5.2xlarge	3.2.4	S	S	S	2,5	10.0
cache.m5.4xlarge	3.2.4	S	S	S	5,0	10.0
cache.m5.12xlarge	3.2.4	S	S	S	12	N/D
cache.m5.24xlarge	3.2.4	S	S	S	25	N/D
cache.m4.large	3.2.4	N	N	N	0,45	1.2
cache.m4.xlarge	3.2.4	S	N	N	0.75	2.8
cache.m4.2xlarge	3.2.4	S	S	S	1,0	10.0

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Multiplexação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m4.4xlarge	3.2.4	S	S	S	2,0	10.0
cache.m4.10xlarge	3.2.4	S	S	S	5,0	10.0
cache.t4g.micro	3.2.4	N	N	N	0,064	5,0
cache.t4g.small	5.0.6	N	N	N	0,128	5,0
cache.t4g.medium	5.0.6	N	N	N	0,256	5,0
cache.t3.micro	3.2.4	N	N	N	0,064	5,0
cache.t3.small	3.2.4	N	N	N	0,128	5,0
cache.t3.medium	3.2.4	N	N	N	0,256	5,0
cache.t2.micro	3.2.4	N	N	N	0,064	1.024
cache.t2.small	3.2.4	N	N	N	0,128	1.024
cache.t2.medium	3.2.4	N	N	N	0,256	1.024

Otimizado para memória

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Multiple ação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r7g.large	6.2	N	N	N	0,937	12,5
cache.r7g.xlarge	6.2	S	S	S	1.876	12,5
cache.r7g.2xlarge	6.2	S	S	S	3,75	15
cache.r7g.4xlarge	6.2	S	S	S	7,5	15
cache.r7g.8xlarge	6.2	S	S	S	15	N/D
cache.r7g.12xlarge	6.2	S	S	S	22,5	N/D
cache.r7g.16xlarge	6.2	S	S	S	30	N/D
cache.r6g.large	5.0.6	N	N	N	0.75	10.0
cache.r6g.xlarge	5.0.6	S	S	S	1,25	10.0
cache.r6g.2xlarge	5.0.6	S	S	S	2,5	10.0
cache.r6g.4xlarge	5.0.6	S	S	S	5,0	10.0

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Múltipla aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6g.8xlarge	5.0.6	S	S	S	12	N/D
cache.r6g.12xlarge	5.0.6	S	S	S	20	N/D
cache.r6g.16xlarge	5.0.6	S	S	S	25	N/D
cache.r5.large	3.2.4	N	N	N	0.75	10.0
cache.r5.xlarge	3.2.4	S	N	N	1,25	10.0
cache.r5.2xlarge	3.2.4	S	S	S	2,5	10.0
cache.r5.4xlarge	3.2.4	S	S	S	5,0	10.0
cache.r5.12xlarge	3.2.4	S	S	S	12	N/D
cache.r5.24xlarge	3.2.4	S	S	S	25	N/D
cache.r4.large	3.2.4	N	N	N	0.75	10.0
cache.r4.xlarge	3.2.4	S	N	N	1,25	10.0
cache.r4.2xlarge	3.2.4	S	S	S	2,5	10.0
cache.r4.4xlarge	3.2.4	S	S	S	5,0	10.0

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Múltipla ação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r4.8xlarge	3.2.4	S	S	S	12	N/D
cache.r4.16xlarge	3.2.4	S	S	S	25	N/D

Otimizada para memória com classificação de dados em níveis

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Múltipla ação aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6gd.xlarge	6.2.0	S	N	N	1,25	10
cache.r6gd.2xlarge	6.2.0	S	S	S	2,5	10
cache.r6gd.4xlarge	6.2.0	S	S	S	5,0	10

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Múltipla aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6gd.8xlarge	6.2.0	S	S	S	12	N/D
cache.r6gd.12xlarge	6.2.0	S	S	S	20	N/D
cache.r6gd.16xlarge	6.2.0	S	S	S	25	N/D

Otimizadas para rede

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Múltipla aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.c7gn.large	6.2	N	N	N	6,25	30
cache.c7gn.xlarge	6.2	S	S	S	12,5	40

Tipo de instância	Versão mínima compatível do Redis OSS	Aprimorado I/O com Redis OSS 5.0.6+	Descarregamento de TLS com Redis OSS 6.2.5+	I/O Múltipla aprimorada com Redis OSS 7.0.4+	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.c7g.n.2xlarge	6.2	S	S	S	25	50
cache.c7g.n.4xlarge	6.2	S	S	S	50	N/D
cache.c7g.n.8xlarge	6.2	S	S	S	100	N/D
cache.c7g.n.12xlarge	6.2	S	S	S	150	N/D
cache.c7g.n.16xlarge	6.2	S	S	S	200	N/D

Tipos de nós compatíveis por região da AWS

Os tipos de nós compatíveis podem variar entre AWS as regiões. Para obter mais detalhes, consulte os [ElastiCache preços da Amazon](#).

Instâncias de desempenho com capacidade de intermitência

Você pode iniciar nós de cache T4G, T3-Standard e T2-Standard de uso geral com capacidade de intermitência na Amazon. ElastiCache Esses nós fornecem um nível de linha de base de desempenho de CPU, além da capacidade de intermitência do uso de CPU a qualquer momento até o consumo dos créditos acumulados. Um crédito de CPU oferece o desempenho de um núcleo de CPU completo por um minuto.

ElastiCacheOs nós T4G, T3 e T2 da Amazon são configurados como padrão e adequados para cargas de trabalho com uma utilização média de CPU consistentemente abaixo do desempenho básico da instância. Para intermitências acima da linha de base, o nó gasta os créditos acumulados no seu saldo de créditos de CPU. Se o nó estiver em execução abaixo dos créditos acumulados, o desempenho será gradualmente reduzido para o nível de desempenho de linha de base. Essa redução gradual garante que o nó não experimente uma queda acentuada de desempenho quando seu saldo de créditos acumulados de CPU for esgotado. Para obter mais informações, consulte [Créditos de CPU e desempenho básico para instâncias de desempenho intermitentes no Guia EC2](#) do usuário da Amazon.

A tabela a seguir lista os tipos de nó de desempenho com capacidade de intermitência, a taxa na qual os créditos de CPU são obtidos por hora. Também mostra o número máximo de créditos de CPU ganhos que um nó pode acumular e o número de v CPUs por nó. Além disso, fornece o nível de desempenho de linha de base como uma porcentagem do desempenho total de um núcleo (usando uma única vCPU).

Créditos de CPU ganhos por hora	Máximo de créditos obtidos que podem ser acumulados*	v CPUs	Desempenho de linha de base por vCPU	Memória (GiB)	desempenho de rede
12	288	2	10%	0,5	Até 5 gigabits
24	576	2	20%	1,37	Até 5 gigabits
24	576	2	20%	3,09	Até 5 gigabits
12	288	2	10%	0,5	Até 5 gigabits
24	576	2	20%	1,37	Até 5 gigabits

Créditos de CPU ganhos por hora	Máximo de créditos obtidos que podem ser acumulados*	v CPUs	Desempenho de linha de base por vCPU	Memória (GiB)	desempenho de rede
24	576	2	20%	3,09	Até 5 gigabits
6	144	1	10%	0,5	Baixo a moderado
12	288	1	20%	1,55	Baixo a moderado
24	576	2	20%	3.22	Baixo a moderado

* O número de créditos que podem ser acumulados é equivalente ao número de créditos que podem ser obtidos em um período de 24 horas.

** O desempenho de linha de base na tabela é por vCPU. Alguns tamanhos de nós que têm mais de uma vCPU. Para isso, calcule a utilização básica da CPU para o nó multiplicando a porcentagem de vCPU pelo número de v. CPUs

As seguintes métricas de crédito de CPU estão disponíveis para instâncias expansíveis T3 e T4g:

 Note

Essas métricas não estão disponíveis para instâncias expansíveis T2.

- CPUCreditUsage
- CPUCreditBalance

Para obter mais informações sobre essas métricas, consulte [Métricas de crédito de CPU](#).

Além disso, saiba estes detalhes:

- Todos os tipos de nós da geração atual são criados em uma Virtual Private Cloud (VPC) com base na Amazon VPC por padrão.
- Os arquivos somente de acréscimo (AOF) do Redis OSS não são compatíveis com instâncias T2. As variáveis de configuração do Redis OSS `appendonly` e `appendfsync` não são compatíveis com o Redis OSS versão 2.8.22 e posteriores.

Informações relacionadas

- [Características e detalhes ElastiCache do produto Amazon](#)
- [Parâmetros específicos do tipo de nó do Memcached para o Memcached](#)
- [Parâmetros do Valkey e do Redis OSS](#)
- [Criptografia em trânsito \(TLS\)](#)

Reinicialização de nós

Algumas alterações exigem que um cluster do Valkey, Memcached ou Redis OSS seja reinicializado para que as alterações sejam aplicadas. Por exemplo, para alguns parâmetros, a alteração do valor do parâmetro em um `parameter group` é aplicada somente após uma reinicialização.

Tópicos

- [Reinicialização de nós do Redis OSS \(somente para o modo cluster desabilitado\)](#)
- [Reinicialização de um cluster para Memcached](#)

Reinicialização de nós do Redis OSS (somente para o modo cluster desabilitado)

Para clusters do Valkey ou Redis OSS (modo cluster desabilitado), os parâmetros nos grupos de parâmetros que são aplicados somente após a reinicialização são:

- `activeresharding`
- bancos de dados

Os nós do Redis só podem ser atualizados por meio do console do ElastiCache. Você só pode reinicializar um único nó de cada vez. Para reinicializar vários nós, você deve repetir o processo para cada nó.

i Alterações de parâmetro do Valkey ou Redis OSS (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster do Valkey ou Redis OSS (modo cluster habilitado), siga as etapas a seguir.

- activerehashing
 - bancos de dados
1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
 2. Exclua o cluster do Valkey ou Redis OSS (modo cluster habilitado). Consulte [Excluindo um cluster no ElastiCache](#).
 3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Utilização do Console de gerenciamento da AWS

Você pode reinicializar um nó usando o console do ElastiCache.

Para reinicializar um nó (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS que se aplica.
3. No painel de navegação esquerdo, escolha Redis OSS.

É exibida uma lista de clusters que executam o Redis OSS.
4. Escolha o cluster em Cluster Name (Nome do cluster).
5. Em Node name (Nome do nó), escolha o botão de opção ao lado do nó que você deseja reinicializar.
6. Escolha Actions (Ações) e, em seguida, Reboot node (Reinicializar nó).

Para reinicializar vários nós, repita as etapas de 2 a 5 para cada nó que você deseja reinicializar. Você não precisa esperar que um nó termine a reinicialização para reinicializar outro.

Reinicialização de um cluster para Memcached

Quando você reinicializa um cluster do Memcached, o cluster libera todos os seus dados e reinicializa seu mecanismo. Durante esse processo, você não pode acessar o cluster. Como o cluster liberou todos os seus dados, quando o cluster estiver disponível novamente, você começará com um cluster vazio.

Você pode reinicializar um cluster usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Se você usar o console do ElastiCache, a AWS CLI ou a API do ElastiCache, só poderá iniciar a reinicialização de um único cluster. Para reinicializar vários clusters, você deve repetir o processo ou a operação.

Utilização do Console de gerenciamento da AWS

Você pode reinicializar um cluster usando o console do ElastiCache.

Para reinicializar um cluster (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS de interesse.
3. No painel de navegação, escolha o mecanismo em execução no cluster que deseja reinicializar.

É exibida uma lista de clusters que executam o mecanismo escolhido.

4. Escolha o cluster a ser reinicializado marcando a caixa à esquerda do nome do cluster.

O botão Reinicializar é ativado.

Se você escolher mais de um cluster, o botão Reinicializar não estará ativo.

5. Escolha Reboot.

É exibida a tela de confirmação de reinicialização do cluster.

6. Para reiniciar o cluster, escolha Reboot. O status do cluster mudará para reinicializando nós de cluster.

Para não reinicializar o cluster, escolha Cancel.

Para reinicializar vários clusters, repita as etapas de 2 a 5 para cada cluster que você deseja reinicializar. Você não precisa esperar que um cluster termine a reinicialização para reinicializar outro.

Para reinicializar um nó específico, selecione o nó e escolha Reboot (Reinicializar).

Como usar o AWS CLI

Para reinicializar um cluster (AWS CLI), use a operação da CLI `reboot-cache-cluster`.

Para reinicializar nós específicos no cluster, use o `--cache-node-ids-to-reboot` para listar os clusters específicos para reinicializar. O comando a seguir reinicializa os nós 0001, 0002 e 0004 de `my-cluster`.

Para Linux, macOS ou Unix:

```
aws elasticache reboot-cache-cluster \
  --cache-cluster-id my-cluster \
  --cache-node-ids-to-reboot 0001 0002 0004
```

Para Windows:

```
aws elasticache reboot-cache-cluster ^
  --cache-cluster-id my-cluster ^
  --cache-node-ids-to-reboot 0001 0002 0004
```

Para reinicializar todos os nós no cluster, use o parâmetro `--cache-node-ids-to-reboot` e liste todos os ID de nós do cluster. Para obter mais informações, consulte [reboot-cache-cluster](#).

Uso da API do ElastiCache

Para reinicializar um cluster usando a API do ElastiCache, use a ação `RebootCacheCluster`.

Para reinicializar nós específicos no cluster, use o `CacheNodeIdsToReboot` para listar os clusters específicos para reinicializar. O comando a seguir reinicializa os nós 0001, 0002 e 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>
```

Para reinicializar todos os nós no cluster, use o parâmetro `CacheNodeIdsToReboot` e liste todos os ID de nós do cluster. Para obter mais informações, consulte [RebootCacheCluster](#).

Substituição de nós (Valkey e Redis OSS)

O Amazon ElastiCache atualiza frequentemente sua frota com patches e atualizações aplicados às instâncias de forma contínua. No entanto, de tempos em tempos, precisamos executar novamente seus nós do ElastiCache para aplicar atualizações obrigatórias do sistema operacional ao host subjacente. Essas substituições são necessárias para aplicar atualizações que fortalecem a segurança, a confiabilidade e o desempenho operacional.

Você tem a opção de gerenciar essas substituições a qualquer momento antes da janela agendada para a substituição do nó. Ao gerenciar uma substituição sozinho, sua instância recebe a atualização do sistema operacional quando você executa novamente o nó e a substituição de nó agendada é cancelada. Você pode continuar recebendo alertas que indicam que a substituição do nó ocorrerá. Caso já tenha atenuado manualmente a necessidade da manutenção, você pode ignorar esses alertas.

Note

Os nós de cache de substituição gerados automaticamente pelo Amazon ElastiCache podem ter endereços IP diferentes. Você é responsável por revisar a configuração do aplicativo para garantir que os nós de cache estejam associados aos endereços IP apropriados.

A lista a seguir identifica as ações que você pode realizar quando o ElastiCache programa um de seus nós do Valkey ou Redis OSS para substituição. Para agilizar a busca das informações de que você precisa para sua situação, escolha no menu a seguir.

- [Do nothing](#): deixar que o Amazon ElastiCache substitua o nó conforme programado.
- [Change your maintenance window](#): alterar sua janela de manutenção para um melhor momento.
- Configurações do Valkey ou Redis OSS (modo cluster habilitado)
 - [Replace the only node in any Valkey or Redis OSS cluster](#): um procedimento para substituir um nó em um cluster do Valkey ou Redis OSS usando backup e restauração.
 - [Replace a replica node in any Valkey or Redis OSS cluster](#): um procedimento para substituir uma réplica de leitura em qualquer cluster do Valkey ou Redis OSS ao aumentar e diminuir a contagem de réplicas sem tempo de inatividade do cluster.
 - [Replace any node in a Valkey or Redis OSS \(cluster mode enabled\) shard](#): um procedimento dinâmico sem tempo de inatividade do cluster para substituir um nó em um cluster do Valkey ou Redis OSS (modo cluster habilitado) com aumento ou redução de escala na horizontal.

- Configurações do Valkey ou Redis OSS (modo cluster desabilitado)
 - [Replace the only node in any Valkey or Redis OSS cluster](#): procedimento para substituir qualquer nó em um cluster do Valkey ou Redis OSS usando backup e restauração.
 - [Replace a replica node in any Valkey or Redis OSS cluster](#): um procedimento para substituir uma réplica de leitura em qualquer cluster do Valkey ou Redis OSS ao aumentar e diminuir a contagem de réplicas sem tempo de inatividade do cluster.
 - [Replace a node in a Valkey or Redis OSS \(cluster mode disabled\) cluster](#): procedimento para substituir um nó em um cluster do Valkey ou Redis OSS (modo cluster desabilitado) usando replicação.
 - [Replace a Valkey or Redis OSS \(cluster mode disabled\) read-replica](#): um procedimento para substituir manualmente uma réplica de leitura em um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado).
 - [Replace a Valkey or Redis OSS \(cluster mode disabled\) primary node](#): um procedimento para substituir manualmente o nó primário em um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado).

Opções de substituição de nós do Valkey e Redis OSS

- Não fazer nada: se você não fizer nada, o ElastiCache substituirá o nó conforme programado.

Para configurações sem cluster com failover automático habilitado, os clusters no Valkey 7.2 e posteriores e no Redis OSS 5.0.6 e posteriores concluem a substituição enquanto o cluster permanece on-line e atende às solicitações de gravação recebidas. Para clusters habilitados para failover automático no Redis OSS 4.0.10 ou anteriores, você poderá observar uma breve interrupção da gravação, de até alguns segundos, associada à atualização do DNS.

Se o nó for membro de um cluster habilitado para failover automático, o ElastiCache para Valkey ou Redis OSS fornecerá maior disponibilidade durante a aplicação de patches, as atualizações e outras substituições de nós relacionadas à manutenção.

As configurações de cluster ElastiCache que estão definidas para usar clientes de cluster ElastiCache para Valkey ou Redis OSS agora são concluídas enquanto o cluster atende às solicitações de gravação recebidas.

Para configurações sem cluster com failover automático habilitado, os clusters no Valkey 7.2 e posteriores e no Redis OSS 5.0.6 e posteriores concluem a substituição enquanto o cluster permanece on-line e atende às solicitações de gravação recebidas. Para clusters habilitados para failover automático no Redis OSS 4.0.10 ou anteriores, você poderá observar uma breve interrupção da gravação, de até alguns segundos, associada à atualização do DNS.

Se o nó for autônomo, o Amazon ElastiCache primeiro ativará um nó de substituição e depois fará a sincronização a partir do nó existente. O nó existente não está disponível para solicitações de serviço durante esse período. Uma vez concluída a sincronização, o nó existente é encerrado, e o novo nó assume seu lugar. O ElastiCache faz o melhor esforço possível para reter seus dados durante esta operação.

- Mudar sua janela de manutenção: para eventos de manutenção programada, você recebe um email ou uma notificação do evento do ElastiCache. Nesses casos, se você mudar sua janela de manutenção antes da hora de substituição programada, o nó será substituído no novo horário. Para obter mais informações, consulte:
 - [Modificação de um cluster do ElastiCache](#)
 - [Modificação de um grupo de replicação](#)

 Note

A capacidade de alterar sua janela de substituição movendo sua janela de manutenção só está disponível quando a notificação do ElastiCache inclui uma janela de manutenção. Se a notificação não inclui uma janela de manutenção, não é possível alterar a janela de substituição.

Por exemplo, digamos que seja quinta-feira, 9 de novembro, às 15h e a próxima janela de manutenção seja sexta-feira, 10 de novembro, às 17h. Veja estes três cenários e seus resultados:

- Você altera sua janela de manutenção para sexta-feira, 16h (após a data e hora atual e antes da próxima janela de manutenção programada). O nó é substituído na sexta-feira, 10 de novembro, às 16h.

- Você altera sua janela de manutenção para sábado, 16h (após a data e hora atual e a próxima janela de manutenção programada). O nó é substituído no sábado, 11 de novembro, às 16h.
- Você altera sua janela de manutenção para quarta-feira, 16h (dia da semana anterior à data e hora atual). O nó é substituído na próxima quarta-feira, 15 de novembro, às 16h.

Para instruções, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

- Substituir o único nó em qualquer cluster do Valkey ou Redis OSS: caso o cluster não tenha réplicas de leitura, você pode usar o procedimento a seguir para substituir o nó.

Para substituir o único nó usando backup e restauração

1. Crie um snapshot do cluster do nó. Para instruções, consulte [Realização de backups manuais](#).
 2. Crie um novo cluster propagando-o a partir do snapshot. Para instruções, consulte [Restauração de um backup para um novo cache](#).
 3. Exclua o cluster com o nó programado para substituição. Para instruções, consulte [Excluindo um cluster no ElastiCache](#).
 4. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
- Substituir um nó de réplica em qualquer cluster do Valkey ou Redis OSS: para substituir um cluster de réplica, aumente sua contagem de réplicas. Para fazer isso, adicione uma réplica e diminua a contagem de réplicas removendo a réplica que deseja substituir. Esse processo é dinâmico e não tem tempo de inatividade do cluster.

 Note

Caso o fragmento ou o grupo de replicação já tenha cinco réplicas, reverta as etapas 1 e 2.

Para substituir uma réplica em qualquer cluster do Valkey ou Redis OSS

1. Aumente a contagem de réplicas adicionando uma réplica ao estilhaço ou ao grupo de replicação. Para obter mais informações, consulte [Aumento do número de réplicas em um fragmento](#).

2. Exclua a réplica que você deseja substituir. Para obter mais informações, consulte [Diminuição do número de réplicas em um fragmento](#).
 3. Atualize os endpoints no aplicativo.
- Substituir qualquer nó em um fragmento do Valkey ou Redis OSS (modo cluster habilitado): para substituir o nó em um cluster sem nenhum tempo de inatividade, use a refragmentação on-line. Primeiro adicione um estilhaço por expansão e, depois, exclua o estilhaço com o nó a ser substituído por redução.

Para substituir qualquer nó em um cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Aumentar: adicione mais um estilhaço com a mesma configuração que o estilhaço existente que contém o nó a ser substituído. Para obter mais informações, consulte [Adição de fragmentos com refragmentação online](#).
 2. Diminuir: exclua o estilhaço com o nó a ser substituído. Para obter mais informações, consulte [Remoção de fragmentos com refragmentação online](#).
 3. Atualize os endpoints no aplicativo.
- Substituir um nó em um cluster do Valkey ou Redis OSS (modo cluster desabilitado): se o cluster for um cluster do Valkey ou Redis OSS (modo cluster desabilitado) sem nenhuma réplica de leitura, use o procedimento a seguir para substituir o nó.

Como substituir o nó usando a replicação (somente para o modo de cluster desabilitado)

1. Adicione a replicação ao cluster com o nó programado para substituição como primário. Não habilite o recurso Multi-AZ nesse cluster. Para instruções, consulte [Para adicionar replicação a um cluster do Valkey ou Redis OSS sem fragmentos](#).
2. Adicione uma réplica de leitura ao cluster. Para instruções, consulte [Para adicionar nós a um ElastiCache cluster \(console\)](#).
3. Promova a réplica de leitura recém-criada para primário. Para instruções, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\)](#).

4. Exclua o nó agendado para substituição. Para instruções, consulte [Removendo nós de um ElastiCache cluster](#).
 5. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
- Substituir uma réplica de leitura do Valkey ou Redis OSS (modo cluster desabilitado): se o nó for uma réplica de leitura, substitua o nó.

Se o cluster tiver apenas um nó de réplica e o recurso Multi-AZ estiver habilitado, você deverá desabilitar o Multi-AZ antes de excluir a réplica. Para instruções, consulte [Modificação de um grupo de replicação](#).

Para substituir uma réplica de leitura do Valkey ou Redis OSS (modo cluster desabilitado)

1. Exclua a réplica programada para substituição. Para instruções, consulte:
 - [Diminuição do número de réplicas em um fragmento](#)
 - [Removendo nós de um ElastiCache cluster](#)
 2. Adicione uma nova réplica para substituir a que está programada para substituição. Se você usar o mesmo nome que a réplica que acabou de excluir, ignore a etapa 3. Para instruções, consulte:
 - [Aumento do número de réplicas em um fragmento](#)
 - [Adição de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#)
 3. No seu aplicativo, substitua o endpoint da réplica antiga pelo endpoint da nova réplica.
 4. Se você tiver desabilitado o recurso Multi-AZ no início, reabilite-o agora. Para instruções, consulte [Habilitar Multi-AZ](#).
- Substituir um nó primário do Valkey ou Redis OSS (modo cluster desabilitado): se o nó for o nó primário, primeiro promova uma réplica de leitura para primário. Depois, exclua a réplica usada para ser o nó principal.

Se o cluster tiver apenas uma réplica e o recurso Multi-AZ estiver habilitado, você deverá desabilitar o Multi-AZ antes de excluir a réplica na etapa 2. Para instruções, consulte [Modificação de um grupo de replicação](#).

Para substituir um nó primário do Valkey ou Redis OSS (modo cluster desabilitado)

1. Promover uma réplica de leitura para primário. Para instruções, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\)](#).
2. Exclua o nó programado para substituição (o primário antigo). Para instruções, consulte [Removendo nós de um ElastiCache cluster](#).
3. Adicione uma nova réplica para substituir a programada para substituição. Se você usar o mesmo nome que o nó que acabou de excluir, poderá ignorar a alteração de endpoints no seu aplicativo.

Para instruções, consulte [Adição de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#).

4. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
5. Se você tiver desabilitado o recurso Multi-AZ no início, reabilite-o agora. Para instruções, consulte [Habilitar Multi-AZ](#).

Substituição de nós (Memcached)

O Amazon ElastiCache for Memcached atualiza frequentemente sua frota com patches e atualizações aplicados às instâncias de forma contínua. No entanto, de tempos em tempos, precisamos executar novamente seus nós do ElastiCache for Memcached para aplicar atualizações obrigatórias do sistema operacional ao host subjacente. Essas substituições são necessárias para aplicar atualizações que fortalecem a segurança, a confiabilidade e o desempenho operacional.

Você tem a opção de gerenciar essas substituições a qualquer momento antes da janela agendada para a substituição do nó. Ao gerenciar uma substituição sozinho, sua instância recebe a atualização do sistema operacional quando você executa novamente o nó e a substituição de nó agendada é cancelada. Você pode continuar recebendo alertas que indicam que a substituição do nó ocorrerá. Caso já tenha atenuado manualmente a necessidade da manutenção, você pode ignorar esses alertas.

Note

Os nós de cache de substituição gerados automaticamente pelo Amazon ElastiCache podem ter endereços IP diferentes. Você é responsável por revisar a configuração do aplicativo para garantir que os nós de cache estejam associados aos endereços IP apropriados.

A lista a seguir identifica as ações que você pode realizar quando o ElastiCache programa um de seus nós do Memcached para substituição.

- Não fazer nada: se você não fizer nada, o ElastiCache substituirá o nó conforme programado. Quando o ElastiCache substitui automaticamente o nó por um novo nó, o novo nó fica inicialmente vazio.
- Mudar sua janela de manutenção para eventos de manutenção programada, você recebe um email ou uma notificação do evento do ElastiCache. Nesse caso, se você mudar sua janela de manutenção antes da hora de substituição programada, o nó será substituído no novo horário. Para obter mais informações, consulte [Modificação de um cluster do ElastiCache](#).

Note

A capacidade de alterar sua janela de substituição movendo sua janela de manutenção só está disponível quando a notificação do ElastiCache inclui uma janela de manutenção.

Se a notificação não inclui uma janela de manutenção, não é possível alterar a janela de substituição.

Por exemplo, digamos que seja quinta-feira, 9 de novembro, às 15h e a próxima janela de manutenção seja sexta-feira, 10 de novembro, às 17h. Veja estes três cenários e seus resultados:

- Você altera sua janela de manutenção para sexta-feira, 16h (após a data e hora atual e antes da próxima janela de manutenção programada). O nó é substituído na sexta-feira, 10 de novembro, às 16h.
- Você altera sua janela de manutenção para sábado, 16h (após a data e hora atual e a próxima janela de manutenção programada). O nó é substituído no sábado, 11 de novembro, às 16h.
- Você altera sua janela de manutenção para quarta-feira, 16h (dia da semana anterior à data e hora atual). O nó é substituído na próxima quarta-feira, 15 de novembro, às 16h.

Para instruções, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

- Substituir manualmente o nó se você precisar substituir o nó antes da próxima janela de manutenção, substitua o nó manualmente.

Se você substituir o nó manualmente, as chaves serão redistribuídas. Essa redistribuição causa perdas no cache.

Para substituir manualmente um nó Memcached

1. Exclua o nó agendado para substituição. Para instruções, consulte [Removendo nós de um ElastiCache cluster](#).
2. Adicione um novo nó ao cluster. Para instruções, consulte [Adicionar nós a um ElastiCache cluster](#).
3. Se você não estiver usando a descoberta automática neste cluster, consulte a aplicação e substitua todas as instâncias do endpoint do nó anterior pelo endpoint do novo nó.

Nós reservados

Reservar um ou mais nós do ElastiCache pode ser uma maneira de reduzir custos. Para nós reservados, é necessário pagar uma taxa adiantada que depende do tipo de nó e da duração da reserva, um ou três anos.

Para ver se os nós reservados são uma economia para seus casos de uso, primeiro determine o tamanho do nó e o número de nós necessários. Depois, estime o uso do nó e compare o custo total de usar nós sob demanda e nós reservados. Você pode combinar e misturar o uso de nós reservados e sob demanda nos seus clusters. Para obter informações de preços, consulte [Definição de preço do Amazon ElastiCache](#).

Tópicos

- [Gerenciamento de custos com nós reservados](#)
- [Ofertas de nós reservados padrão](#)
- [Tamanho de nós reservados flexíveis](#)
- [Excluir um nó reservado](#)
- [Ofertas de nós reservados herdados](#)
- [Obtenção de informações sobre ofertas de nós reservados](#)
- [Compra de um nó reservado](#)
- [Obtenção de informações sobre seus nós reservados](#)

Gerenciamento de custos com nós reservados

Reservar um ou mais nós pode ser uma maneira de reduzir custos. Para nós reservados, é necessário pagar uma taxa adiantada que depende do tipo de nó e da duração da reserva, um ou três anos. Essa cobrança é muito menor do que a cobrança de uso por hora dos nós sob demanda.

Para ver se os nós reservados são uma economia para seus casos de uso, primeiro determine o tamanho do nó e o número de nós necessários. Depois, estime o uso do nó e compare o custo total de usar nós sob demanda e nós reservados. Você pode combinar e misturar o uso de nós reservados e sob demanda nos seus clusters. Para obter informações de preços, consulte [Definição de preço do Amazon ElastiCache](#).

A região da AWS, o tipo de nó e o tempo de duração devem ser escolhidos no ato da compra e não podem ser alterados posteriormente.

Você pode usar o Console de gerenciamento da AWS, a AWS CLI ou a API do ElastiCache para listar e comprar ofertas de nós reservados disponíveis.

Para obter mais informações sobre nós reservados, consulte [Nós reservados do Amazon ElastiCache](#).

Ofertas de nós reservados padrão

Ao comprar uma instância de nó reservado (RI) no Amazon ElastiCache, você pode adquirir um compromisso de obter uma taxa com desconto sobre um tipo de instância de nó e região da AWS específicos para a duração da instância de nó reservado. Para usar uma instância de Nós Reservados do Amazon ElastiCache, uma nova instância do nó do ElastiCache deve ser criada, como faria para uma instância sob demanda.

Se as especificações da nova instância de nó reservado corresponderem às de uma instância de nó reservado existente para a sua conta, será cobrada a taxa de desconto oferecida para a instância de nó reservado. Caso contrário, uma taxa sob demanda será cobrada para a instância do nó. Esses IRs padrão estão disponíveis a partir das famílias de instâncias R5 e M5.

Note

Todos os tipos de ofertas discutidos a seguir estão disponíveis para períodos de vigência de um e três anos.

Tipos de oferta

Sem adiantamento O RI fornece acesso a uma instância ElastiCache reservado sem a necessidade de pagamento adiantado. Sua instância do ElastiCache reservado sem adiantamento cobra uma taxa por hora com desconto a cada hora dentro do prazo, independentemente do uso.

Adiantamento parcial A RI exige que uma parte da instância reservada do ElastiCache seja paga antecipadamente. As horas restantes do período de vigência serão cobradas com base em uma taxa horária com desconto, independentemente do uso. Essa opção é a substituição da opção antiga de utilização pesada, que é explicada na próxima seção.

A RI adiantamento integral exige que o pagamento integral seja feito no início do período de vigência da RI. Você não incorrer em outros custos pelo restante do período, independentemente do número de horas usadas.

Tamanho de nós reservados flexíveis

Todos os nós reservados têm tamanho flexível. Ao adquirir um nó reservado, uma das especificações feitas é o tipo de nó, por exemplo, `cache.r6g.xlarge`. Para obter mais informações sobre tipos de nós, consulte [Definição de preços do Amazon ElastiCache](#).

Se você tiver um nó e precisar escalá-lo para uma capacidade maior, o nó reservado será automaticamente aplicado ao nó escalado. Ou seja, seus nós reservados são automaticamente aplicados ao uso de qualquer tamanho na mesma família de nós. Os nós reservados de tamanho flexível estão disponíveis para nós com a mesma região da AWS. Nós reservados de tamanho flexível só podem reduzir a escala horizontalmente em suas famílias de nós. Por exemplo, um nó reservado para um `cache.r6g.xlarge` pode ser aplicado a um `cache.r6g.2xlarge`, mas não a um `cache.r6gd.large`, porque `cache.r6g` e `cache.r6gd` são famílias de nós diferentes.

Flexibilidade de tamanho significa que você pode se mover livremente entre configurações dentro da mesma família de nós. Por exemplo, você pode passar de um nó reservado `r6g.xlarge` (8 unidades normalizadas) para dois nós reservados `r6g.large` (8 unidades normalizadas) ($2 \times 4 = 8$ unidades normalizadas) na mesma região da AWS sem custo adicional.

Atualização de nós do Redis OSS para o Valkey

Com o lançamento do Valkey no ElastiCache, agora você pode aplicar seu desconto de nós reservados do Redis OSS ao mecanismo de cache Valkey. Você pode realizar o upgrade do Redis OSS para o Valkey e ainda se beneficiar dos contratos e reservas existentes. Além de poder aplicar seus benefícios no mecanismo e família de nós de cache, você pode até receber mais valor incremental. O Valkey tem um preço de 20% de desconto em relação ao Redis OSS e, com a flexibilidade de nós reservados, você pode usar seus nós reservados do Redis OSS para cobrir 20% mais nós em execução do Valkey.

Para calcular a taxa de desconto, cada combinação de nó e mecanismo do ElastiCache tem um fator de normalização que é medido em unidades. As unidades de nós reservados podem ser aplicadas a qualquer nó em execução dentro da família de instâncias do nó reservado para um determinado mecanismo. Os nós reservados do Redis OSS podem ser aplicados adicionalmente em todos os mecanismos para cobrir a execução dos nós do Valkey. Como o Valkey tem um preço com desconto em relação ao Redis OSS e ao Memcached, suas unidades para um determinado tipo de instância são mais baixas, o que permite que um nó reservado do Redis OSS cubra mais nós do Valkey.

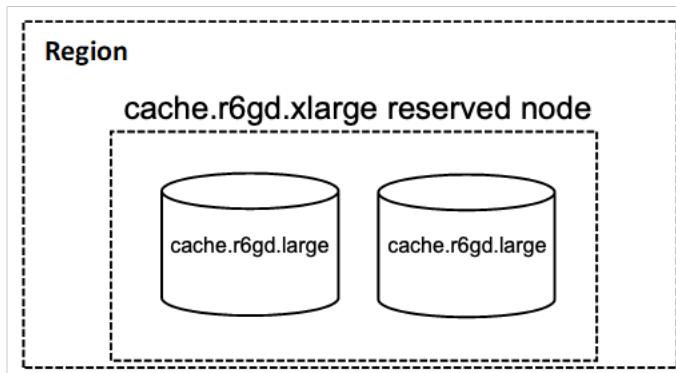
Como exemplo, digamos que você tenha comprado um nó reservado para um `cache.r7g.4xlarge` para o mecanismo Redis OSS (32 unidades) e esteja executando um nó `cache.r7g.4xlarge` do Redis OSS (32 unidades). Se você realizar o upgrade do nó para Valkey, o fator de normalização do nó em execução cai para 25,6 unidades, e seu nó reservado existente fornece 6,4 unidades adicionais para usar em qualquer outro nó do Valkey ou Redis OSS em execução na família `cache.r7g` na região. Você pode usar isso para cobrir 25% de outro nó `cache.r7g.4xlarge` do Valkey na conta (25,6 unidades) ou 100% de um nó `cache.r7g.xlarge` do Valkey (6,4 unidades).

Comparação de uso com unidades normalizadas

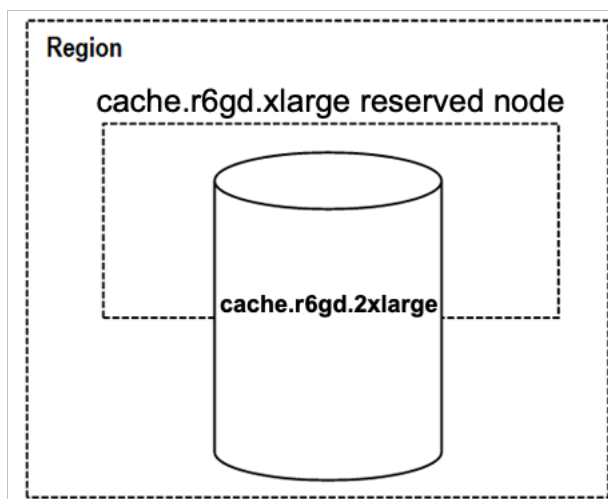
Você pode comparar o uso de diferentes tamanhos de nós reservados usando unidades normalizadas. Por exemplo, uma hora de uso em dois nós `cache.r6g.4xlarge` é equivalente a 16 horas de uso em um `cache.r6g.large`. A tabela a seguir mostra o número de unidades normalizadas para cada tamanho de nó:

Tamanho do nó	Unidades normalizadas com Redis OSS ou Memcached	Unidades normalizadas com o Valkey
micro	0,5	0.4
pequeno	1	8.
médio	2	1.6
grande	4	3.2
xlarge	8	6.4
2xlarge	16	12.8
4xlarge	32	25.6
6xlarge	48	38.4
8xlarge	64	51.2
10xlarge	80	64
12xlarge	96	76.8
16xlarge	128	102.4
24xlarge	192	153.6

Por exemplo, você compra um nó reservado `cache.r6gd.xlarge` e tem dois nós reservados `cache.r6gd.large` em execução na sua conta na mesma região da AWS. Nesse caso, o benefício de faturamento é aplicado integralmente a ambos os nós.



Ou, se você tiver uma instância `cache.r6gd.2xlarge` em execução na sua conta na mesma região da AWS, o benefício de cobrança será aplicado a 50% do uso do nó reservado.



Excluir um nó reservado

Os períodos de vigência de um nó reservado envolvem um compromisso de um ou três anos. Você não pode cancelar um nó reservado. No entanto, você pode excluir um nó coberto por um desconto de nó reservado. O processo de exclusão de um nó coberto por um desconto de nó reservado é o mesmo que o de qualquer outro nó.

Se excluir um nó coberto por um desconto de nó reservado, você poderá iniciar outro nó com especificações compatíveis. Neste caso, você continua recebendo a taxa com desconto durante o período de vigência da reserva (um ou três anos).

Ofertas de nós reservados herdados

Existem três níveis de reservas de nós herdados: utilização pesada, utilização média e utilização leve. Os nós podem ser reservados em qualquer nível de utilização para um ou três anos. O tipo de nó, o nível de utilização e o prazo de reserva impactarão seus custos totais. Verifique as economias

que os nós reservados podem fornecer aos seus negócios, comparando vários modelos antes de comprar nós reservados.

Os nós comprados em um nível ou período de vigência de utilização não podem ser convertidos para um nível ou período de vigência de utilização diferente.

Níveis de utilização

Nós reservados de utilização pesada habilitam cargas de trabalho que possuem uma linha de base consistente de cargas de trabalho de capacidade ou estacionárias. Os nós reservados de utilização pesada exigem um alto compromisso antecipado. Porém, se você planeja executar mais de 79% do período de vigência do nó reservado, poderá obter as maiores economias (até 70% de desconto no preço sob demanda). Com nós reservados de utilização pesada, você paga uma taxa única. Em seguida, isso é seguido de uma taxa horária inferior durante o período de vigência, independentemente de o seu nó estar em execução.

Nós reservados de utilização média são a melhor opção se você planeja usar seus nós reservados durante uma grande parcela de tempo, mas deseja uma taxa única mais baixa ou parar de pagar pelo seu nó ao desligá-lo. Os nós reservados de utilização média são uma opção mais econômica quando você planeja executar mais de 40% do período de vigência de nós reservados. Essa opção pode render até 64% de desconto no preço sob demanda. Com nós reservados de utilização média, você paga uma taxa única um pouco maior que a de nós reservados de utilização leve e recebe taxas de uso por hora mais baixas ao executar um nó.

Nós reservados de utilização leve são ideais para cargas de trabalho periódicas que funcionam apenas algumas horas por dia ou alguns dias por semana. Usando nós reservados de utilização leve, você paga uma taxa única seguida de uma taxa de uso por hora com desconto quando seu nó está sendo executado. Você pode começar a economizar quando seu nó estiver executando mais de 17% do período de vigência do nó reservado. Você pode economizar até 56% nas taxas sob demanda durante todo o período de vigência de seu nó reservado.

Ofertas de nós reservados herdados

Oferta	Custo inicial	Taxa de uso	Vantagem
Utilização pesada	Mais alto	A taxa por hora mais baixa. Aplicada ao período de vigência inteiro, independe	O custo geral mais baixo se você planeja executar seus nós reservados mais de

Oferta	Custo inicial	Taxa de uso	Vantagem
		ntemente de você estar ou não usando o nó reservado.	79% do período de vigência de três anos.
Utilização média	Médio	Taxa de uso por hora cobrada por cada hora em que o nó está sendo executado . Nenhuma cobrança por hora quando o nó não está sendo executado.	Adequada para cargas de trabalho elásticas ou quando você espera um uso moderado, superior a 40% do período de vigência de três anos.
Utilização leve	Menor	Taxa de uso por hora cobrada por cada hora em que o nó está sendo executado . Nenhuma cobrança por hora quando o nó não está sendo executado. Taxas por hora são as mais altas entre todos os tipos de oferta, mas aplicam-se somente quando o nó reservado está sendo executado.	Custo geral mais alto se você planeja executar o tempo todo. Contudo, esse é o custo geral mais baixo se você planejar usar seu nó reservado com pouca frequência, ou seja, mais de cerca de 15% de um período de vigência de três anos.
Uso sob demanda (sem nós reservados)	Nenhum	Taxa horária mais alta. Aplicada sempre que o nó está sendo executado.	Maior custo por hora.

Para obter mais informações, consulte [Definição de preço Amazon ElastiCache](#).

Obtenção de informações sobre ofertas de nós reservados

Antes de comprar nós reservados, você pode obter informações sobre ofertas de nós reservados disponíveis.

Os exemplos a seguir mostram como obter a definição de preço e informações sobre ofertas de nós reservados disponíveis usando o Console de gerenciamento da AWS, a AWS CLI e a API do ElastiCache.

Tópicos

- [Obtenção de informações sobre ofertas de nós reservados \(console\)](#)
- [Obtenção de informações sobre ofertas de nós reservados \(AWS CLI\)](#)
- [Obtenção de informações sobre ofertas de nós reservados \(API do ElastiCache\)](#)

Obtenção de informações sobre ofertas de nós reservados (console)

Para obter definição de preço e outras informações sobre ofertas de clusters reservados disponíveis usando o Console de gerenciamento da AWS, use o procedimento a seguir.

Para obter informações sobre ofertas de nós reservados disponíveis

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione Reserved Nodes.
3. Escolha Purchase Reserved Nodes (Comprar nós reservados).
4. Em Mecanismo, escolha Valkey, Memcached ou Redis OSS.
5. Para determinar as ofertas disponíveis, faça seleções nas seguintes opções:
 - Tipo de nó
 - Prazo
 - Offering Type

Após essas seleções, o custo por nó e o custo total de suas seleções serão exibidos em Reservation details (Detalhes da reserva).

6. Escolha Cancel para evitar a compra desses nós e gerar cobranças.

Obtenção de informações sobre ofertas de nós reservados (AWS CLI)

Para obter preços e outras informações sobre ofertas de nós reservados disponíveis para Valkey ou Redis OSS, digite o seguinte comando em um prompt de comando:

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 31536000,
```

```

    "FixedPrice": X.X,
    "UsagePrice": X.X,
    "ProductDescription": "redis",
    "OfferingType": "No Upfront",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": X.XXX,
        "RecurringChargeFrequency": "Hourly"
      }
    ]
  }
}

```

Para obter preços e outras informações sobre ofertas de nós reservados disponíveis para Memcached, digite o seguinte comando em um prompt de comando:

```

{
  "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"  
            }  
        ]  
    }  
}
```

Para obter mais informações, consulte [describe-reserved-cache-nodes-offerings](#) na AWS CLI Referência.

Obtenção de informações sobre ofertas de nós reservados (API do ElastiCache)

Para obter informações sobre preços e informações sobre ofertas de nós reservados disponíveis, chame a ação `DescribeReservedCacheNodesOfferings`.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodesOfferings  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [DescribeReservedCacheNodesOfferings](#) na Referência da API do ElastiCache.

Compra de um nó reservado

Os exemplos a seguir mostram como comprar uma oferta de nó reservado usando o Console de gerenciamento da AWS, a AWS CLI e a API do ElastiCache.

Important

Seguir os exemplos desta seção gerará cobranças na sua conta da AWS que você não poderá reverter.

Tópicos

- [Compra de um nó reservado \(console\)](#)
- [Compra de um nó reservado \(AWS CLI\)](#)
- [Compra de um nó reservado \(API do ElastiCache\)](#)

Compra de um nó reservado (console)

Este exemplo mostra a compra de uma oferta de nó reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, com um ID de nó reservado de myreservationID.

O procedimento a seguir usa o Console de gerenciamento da AWS para comprar a oferta de nó reservado por ID de oferta.

Para comprar nós reservados

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha o link Reserved nodes (Nós reservados).
3. Escolha o botão Purchase reserved nodes (Comprar nós reservados).
4. Em Mecanismo, escolha Valkey, Memcached ou Redis OSS.
5. Para determinar as ofertas disponíveis, faça seleções nas seguintes opções:
 - Tipo de nó
 - Prazo
 - Offering Type
 - Um Reserved node ID (ID de nó reservado) opcional

Após essas seleções, o custo por nó e o custo total de suas seleções serão exibidos em **Reservation details** (Detalhes da reserva).

6. Escolha Purchase (Comprar).

Compra de um nó reservado (AWS CLI)

O exemplo a seguir mostra a compra de uma oferta de cluster reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, com um ID de nó reservado de myreservationID.

Digite o seguinte comando em um prompt de comando:

Para 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
```

Para Windows:

```
aws elasticache purchase-reserved-cache-nodes-offering ^
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
  --reserved-cache-node-id myreservationID
```

Esse comando retorna uma saída semelhante à seguinte:

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

Para obter mais informações, consulte [purchase-reserved-cache-nodes-offering](#) na AWS CLI Referência.

Compra de um nó reservado (API do ElastiCache)

O exemplo a seguir mostra a compra de uma oferta de nó reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, com um ID de cluster reservado de myreservationID.

Chame a operação `PurchaseReservedCacheNodesOffering` com os seguintes parâmetros:

- `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>
```

Para obter mais informações, consulte [PurchaseReservedCacheNodesOffering](#) na Referência da API do ElastiCache.

Obtenção de informações sobre seus nós reservados

Você pode obter informações sobre os nós reservados que você comprou usando o Console de gerenciamento da AWS, o AWS CLI e a API do ElastiCache.

Tópicos

- [Obtenção de informações sobre seus nós reservados \(console\)](#)
- [Obtenção de informações sobre seus nós reservados \(AWS CLI\)](#)
- [Obtenção de informações seus nós reservados \(API do ElastiCache\)](#)

Obtenção de informações sobre seus nós reservados (console)

O procedimento a seguir descreve como usar o Console de gerenciamento da AWS para obter informações sobre os nós reservados que você comprou.

Para obter informações sobre seus nós reservados comprados

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha o link Reserved nodes (Nós reservados).

Os nós reservados para sua conta aparecem na lista Reserved nodes (Nós reservados). Você pode escolher qualquer um dos nós reservados na lista para ver informações detalhadas sobre o nó reservado no painel de detalhes na parte inferior do console.

Obtenção de informações sobre seus nós reservados (AWS CLI)

Para obter informações sobre nós reservados para sua a conta da AWS, digite o seguinte comando em um prompt de comando:

```
aws elasticache describe-reserved-cache-nodes
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
  "CacheNodeType": "cache.xx.small",
```

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

Para obter mais informações, consulte [describe-reserved-cache-nodes](#) na Referência da AWS CLI.

Obtenção de informações seus nós reservados (API do ElastiCache)

Para obter informações sobre nós reservados para sua conta da AWS, execute a operação `DescribeReservedCacheNodes`.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodes  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [DescribeReservedCacheNodes](#) na Referência da API do ElastiCache.

Migração de nós da geração anterior

Os nós de geração anterior são tipos de nó que estão sendo eliminados gradualmente. Se você não tiver clusters existentes usando um tipo de nó de geração anterior, o ElastiCache não oferece suporte à criação de novos clusters com esse tipo de nó.

Devido à quantidade limitada de tipos de nó de geração anterior, não podemos garantir uma substituição bem-sucedida quando um nó se torna não íntegro em seu(s) cluster(s). Nesse cenário, a disponibilidade do cluster pode ser afetada negativamente.

Recomendamos que você migre seu(s) cluster(s) para um novo tipo de nó para melhor disponibilidade e performance. Para obter um tipo de nó recomendado para migrar, consulte [Caminhos de atualização](#). Para obter uma lista completa dos tipos de nó com suporte e dos tipos de nó da geração anterior no ElastiCache, consulte [Tipos de nó compatíveis](#).

Migração de nós em um cluster do Valkey ou Redis OSS

O procedimento a seguir descreve como migrar seu tipo de nó de cluster do Valkey ou Redis OSS usando o Console do ElastiCache. Durante esse processo, o cluster do Valkey ou Redis OSS continuará a atender solicitações com tempo de inatividade mínimo. Dependendo da configuração do cluster, você pode ver os seguintes tempos de inatividade. As estimativas a seguir podem diferir com base em suas configurações específicas:

- O modo cluster desabilitado (nó único) pode ver aproximadamente 60 segundos, principalmente devido à propagação do DNS.
- O modo de cluster desabilitado (com o nó de réplica) pode ver aproximadamente um segundo para clusters que executam o Valkey 7.2 e posteriores ou o Redis OSS 5.0.6 e posteriores. Todas as versões inferiores podem experimentar aproximadamente 10 segundos.
- O modo cluster habilitado pode ver aproximadamente 1 segundo.

Para modificar um tipo de nó de cluster do Valkey ou Redis OSS usando o console:

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Na lista de clusters, escolha o cluster que você deseja migrar.
4. Escolha Actions (Ações) e Modify (Modificar).
5. Escolha o novo tipo de nó na lista de tipos de nó.
6. Se quiser executar o processo de migração imediatamente, escolha Apply immediately (Aplicar imediatamente). Se Apply immediately (Aplicar imediatamente) não for escolhida, o processo de migração será realizado durante a próxima janela de manutenção do cluster.
7. Escolha Modificar. Se você escolheu Apply immediately na etapa anterior, o status do cluster mudará para modifying. Quando o status mudar para available, a modificação estará completa, e você poderá começar a usar o novo cluster.

Para modificar um tipo de nó de cluster do Valkey ou Redis OSS usando a AWS CLI:

Use a API [modify-replication-group](#) conforme mostrado a seguir:

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group /  
  --replication-group-id my-replication-group /  
  --cache-node-type new-node-type /  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-replication-group ^  
  --cache-node-type new-node-type ^  
  --apply-immediately
```

Nesse cenário, o valor de *new-node-type* é o tipo de nó para o qual você está migrando.

Passando o parâmetro `--apply-immediately`, a atualização será aplicada imediatamente quando o grupo de replicação fizer a transição do status `modifying` (modificando) para `available` (disponível). Se `Apply immediately` (Aplicar imediatamente) não for escolhida, o processo de migração será realizado durante a próxima janela de manutenção do cluster.

 Note

Se você não conseguir modificar o cluster com um erro `InvalidCacheClusterState`, você precisa remover primeiro um nó com falha de restauração.

Correção ou remoção de nós com falha na restauração

O procedimento a seguir descreve como corrigir ou remover nós com falha na restauração pelo cluster do Valkey ou Redis OSS. Para saber mais sobre como nós do ElastiCache entram em um estado de falha na restauração, consulte [Visualizando o status ElastiCache do nó](#). É recomendável primeiramente remover eventuais nós em um estado de falha na restauração e, em seguida, migrar os nós da geração anterior restantes no cluster do ElastiCache para um tipo de nó de uma geração mais nova e, por fim, readicionar o número de nós necessário.

Para remover o nó com falha de restauração (console):

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Na lista de clusters, escolha o cluster do qual você deseja remover um nó.
4. Na lista de fragmentos, escolha o fragmento do qual você deseja remover um nó. Ignore esta etapa se o modo cluster estiver desabilitado para o cluster.
5. Na lista de nós, escolha o nó com um status de `restore-failed`.
6. Escolha Actions (Ações) e, em seguida, escolha Delete node (Excluir nó).

Depois de remover os nós com falha na restauração do cluster do ElastiCache, você poderá migrar para um tipo de geração mais novo. Para obter mais informações, consulte acima em [Migração de nós em um cluster do Valkey ou Redis OSS](#).

Para readicionar nós ao cluster do ElastiCache, consulte [Adicionar nós a um ElastiCache cluster](#).

Migração de nós em um cluster do Memcached

Para migrar o ElastiCache for Memcached para um tipo de nó diferente, você deve criar um novo cluster, que sempre começa vazio, que sua aplicação possa preencher.

Para migrar o tipo de nó do cluster do ElastiCache for Memcached usando o console do ElastiCache:

- Crie um novo cluster com o novo tipo de nó. Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).
- No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para obter mais informações, consulte [Localização de endpoints de um cluster \(console\) \(Memcached\)](#)
- Exclua o cluster antigo. Para obter mais informações, consulte [Excluindo um cluster no ElastiCache](#)

Gerenciamento de clusters no ElastiCache

Um cluster é uma coleção de um ou mais nós de cache, todos os quais executam uma instância do software de mecanismo Valkey, Memcached e Redis OSS. Ao criar um cluster, especifique o mecanismo e a versão para que todos os nós usem.

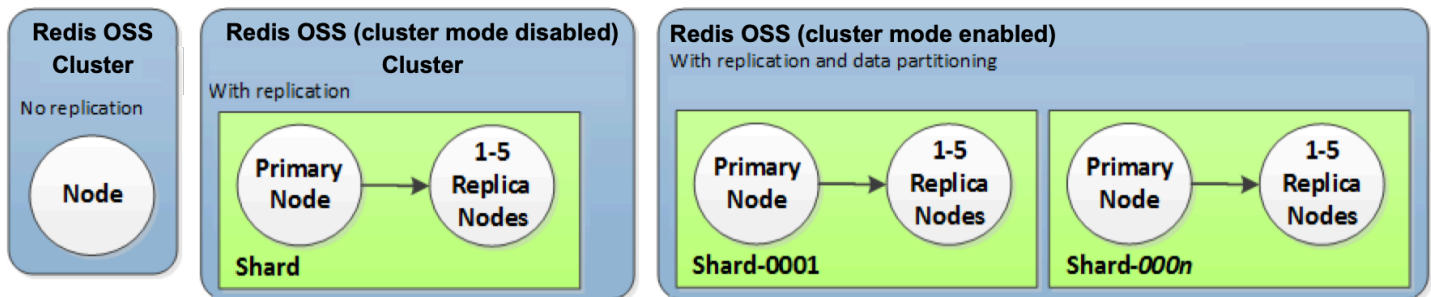
Clusters Valkey e Redis OSS

O diagrama a seguir ilustra um cluster típico do Valkey ou do Redis OSS. Esses clusters podem conter um nó único ou até seis nós dentro de um fragmento (API/CLI: grupo de nós). Um cluster do Valkey ou Redis OSS (modo cluster desabilitado) de nó único não possui fragmentos e um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com vários nós tem um único fragmento. Clusters do Valkey ou Redis OSS (modo cluster habilitado) podem ter até 500 fragmentos, com seus dados particionados nos fragmentos. O limite de nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo for Valkey 7.2 e posteriores ou Redis OSS 5.0.6 e posteriores. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#). Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

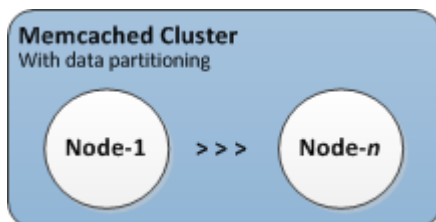
Quando você possui múltiplos nós em um fragmento do Valkey ou Redis OSS, um dos nós é um nó primário de leitura/gravação. Todos os outros nós no estilhaço são réplicas somente leitura.

Os clusters típicos do Valkey ou Redis OSS são semelhantes ao seguinte.



Clusters do Memcached

Clusters típicos do Memcached têm a seguinte aparência. Os clusters do Memcached contêm de 1 a 60 nós em que você particiona horizontalmente seus dados.



Operações do Elasticache para Valkey, Memcached e Redis OSS

A maioria das operações do ElastiCache é realizada no nível do cluster. Você pode configurar um cluster com um número específico de nós e um parameter group que controla as propriedades de cada nó. Todos os nós de um cluster são do mesmo tipo e têm as mesmas configurações de parameter group e security group.

Cada cluster deve ter um identificador de cluster. O identificador de cluster é um nome fornecido pelo cliente para o cluster. Esse identificador especifica um cluster específico ao interagir com os comandos da API do ElastiCache e da AWS CLI. O identificador de cluster deve ser exclusivo para esse cliente em uma região da AWS.

O ElastiCache oferece suporte para várias versões do mecanismo. A menos que você tenha motivos específicos, recomendamos usar a versão mais recente.

Os clusters do ElastiCache foram criados para serem acessados ao usar uma instância do Amazon EC2. Ao executar seu cluster em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC, é possível acessá-lo externamente à AWS. Para obter mais informações, consulte [Acessando ElastiCache recursos de fora AWS](#).

Para obter uma lista das versões compatíveis, consulte [Mecanismos e versões compatíveis](#), [Versões compatíveis do mecanismo do Redis OSS](#) e [Compatível com ElastiCache versões do Memcached](#).

Escolhendo um tipo de rede em ElastiCache

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4 e IPv6), permitindo que você configure seu cluster para aceitar:

- somente IPv4 conexões,
- somente IPv6 conexões,
- ambos IPv4 e IPv6 conexões (pilha dupla)

IPv6 [é compatível com cargas de trabalho usando Valkey 7.2 e versões posteriores, ou Redis OSS 6.2 e versões posteriores, em todas as instâncias criadas no sistema Nitro](#). Não há custos adicionais para acessar ElastiCache IPv6.

Note

A migração de clusters criados antes da disponibilidade de IPV6 /dual-stack não é suportada. A alternância entre tipos de rede em clusters recém-criados também não tem suporte.

IPv6 [é compatível com cargas de trabalho usando o Memcached 1.6.6 em diante em todas as instâncias criadas no sistema Nitro](#). Não há custos adicionais para acessar ElastiCache IPv6.

Configurar sub-redes para o tipo de rede

Se você criar um cluster em uma Amazon VPC, deverá especificar um grupo de sub-redes. ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. ElastiCache os clusters exigem uma sub-rede de pilha dupla com IPv6 endereços atribuídos a eles para operar no modo de pilha dupla e uma IPv6 sub-rede somente para operar como somente. IPv4 IPv6

Usar pilha dupla

Ao usar o ElastiCache Redis OSS no modo de cluster ativado, do ponto de vista de um aplicativo, conectar-se a todos os nós do cluster por meio do endpoint de configuração não é diferente de conectar-se diretamente a um nó de cache individual. Para conseguir isso, um cliente com reconhecimento de cluster deve participar de um processo de descoberta de clusters e solicitar as informações de configuração de todos os nós. O protocolo de descoberta do Redis oferece suporte a apenas um IP por nó.

Ao criar um cluster com ElastiCache for Memcached e escolher dual-stack como o tipo de rede, você precisa designar um tipo de descoberta de IP — ou, IPv4 IPv6 ElastiCache usará como padrão o tipo de rede e a descoberta de IP IPv6, mas isso pode ser alterado. Se você usar a Descoberta automática, somente os endereços IP do tipo de IP escolhido serão retornados ao cliente Memcached. Para obter mais informações, consulte [Identificar automaticamente nós no seu cluster \(Memcached\)](#).

Para manter a compatibilidade retroativa com todos os clientes existentes, é introduzida a descoberta de IP, que permite selecionar o tipo de IP (ou seja, IPv4 ou IPv6) a ser anunciado no protocolo de descoberta. Embora isso limite a descoberta automática a apenas um tipo de IP, a pilha dupla ainda é benéfica para cargas de trabalho habilitadas para o modo de cluster, pois permite migrações (ou reversões) de um IPv4 para um IPv6 tipo de IP de descoberta sem tempo de inatividade.

Clusters de ElastiCache pilha dupla habilitados para TLS

Quando o TLS está habilitado para ElastiCache clusters, as funções de descoberta de `clusterscluster slots`, `como,cluster shards`, e `cluster nodes` com Valkey ou Redis OSS e `config get cluster` com Memcached retornam nomes de host em vez de IPs. Os nomes de host são então usados em vez de se conectar IPs ao ElastiCache cluster e realizar um handshake TLS. Isso significa que os clientes não serão afetados pelo parâmetro de descoberta de IP. Para clusters habilitados para TLS, o parâmetro de descoberta de IP não tem efeito no protocolo IP preferencial. Em vez disso, o protocolo IP usado será determinado pelo protocolo IP que o cliente prefere ao resolver nomes de host DNS.

Para obter exemplos sobre como configurar uma preferência de protocolo IP ao resolver nomes de host DNS, consulte [Clusters do ElastiCache de pilha dupla habilitados para TLS](#).

Usando o Console de gerenciamento da AWS (Valkey e Redis OSS)

Ao criar um cluster usando o Console de gerenciamento da AWS, em Conectividade, escolha um tipo de rede IPv6 ou pilha dupla. IPv4 Se você estiver criando um cluster Valkey ou Redis OSS (modo de cluster ativado) e escolher pilha dupla, deverá selecionar um tipo de IP de descoberta, ou, IPv6 IPv4

Para acessar mais informações, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#) ou [Criação de um cluster do Valkey ou Redis OSS \(modo cluster desabilitado\) \(console\)](#).

Ao criar um grupo de replicação usando o Console de gerenciamento da AWS, escolha um tipo de rede IPv6 ou pilha dupla. IPv4 Se você escolher pilha dupla, deverá selecionar um tipo de IP de descoberta, IPv6 ou IPv4.

Para acessar mais informações, consulte [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) do início](#) ou [Criação de um grupo de replicação no Valkey ou Redis OSS \(modo cluster habilitado\) do início](#).

Usando o Console de gerenciamento da AWS(Memcached)

Ao criar um cluster usando o Console de gerenciamento da AWS, em Conectividade, escolha um tipo de rede IPv6 ou pilha dupla. IPv4 Se você escolher pilha dupla, deverá selecionar um tipo de IP de descoberta, IPv6 ou IPv4.

Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).

Uso da CLI com Valkey, Memcached ou Redis OSS.

Redis OSS

Ao criar um cluster com Valkey ou Redis OSS usando a CLI, você usa o [create-cache-cluster](#) comando e especifica os parâmetros e: NetworkType IPDiscovery

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \
  --cache-cluster-id "cluster-test" \
  --engine redis \
  --cache-node-type cache.m5.large \
  --num-cache-nodes 1 \
  --network-type dual_stack \
  --ip-discovery ipv4
```

Para Windows:

```
aws elasticache create-cache-cluster ^
  --cache-cluster-id "cluster-test" ^
  --engine redis ^
  --cache-node-type cache.m5.large ^
  --num-cache-nodes 1 ^
  --network-type dual_stack ^
  --ip-discovery ipv4
```

Ao criar um grupo de replicação com o modo de cluster desativado usando a CLI, você usa [create-replication-group](#) comando e especifica NetworkType os IPDiscovery parâmetros e:

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --primary-cluster-id redis01 ^  
  --network-type dual_stack ^  
  --ip-discovery ipv4
```

Ao criar um grupo de replicação com o modo de cluster ativado e usado IPv4 para descoberta de IP usando a CLI, você usa [create-replication-group](#) comando e especifica NetworkType os IPDiscovery parâmetros e:

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id demo-cluster \  
  --replication-group-description "demo cluster" \  
  --cache-node-type cache.m5.large \  
  --num-node-groups 2 \  
  --engine redis \  
  --cache-subnet-group-name xyz \  
  --network-type dual_stack \  
  --ip-discovery ipv4 \  
  --region us-east-1
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv4 ^
  --region us-east-1
```

Ao criar um grupo de replicação com o modo de cluster ativado e usado IPv6 para descoberta de IP usando a CLI, você usa [create-replication-group](#) comando e especifica NetworkType os IPDiscovery parâmetros e:

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv6 \
  --region us-east-1
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv6 ^
  --region us-east-1
```

Memcached

Ao criar um cluster com o Memcached usando a CLI, você usa o [create-cache-cluster](#) comando e especifica os parâmetros e: NetworkType IPDiscovery

Para 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
```

Para 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
```

Identificar automaticamente nós no seu cluster (Memcached)

Para clusters que executam o mecanismo Memcached, ElastiCache oferece suporte à descoberta automática — a capacidade dos programas cliente identificarem automaticamente todos os nós em um cluster e iniciarem e manterem conexões com todos esses nós.

Note

A descoberta automática é adicionada para clusters executados no Amazon ElastiCache Memcached. A Descoberta automática não está disponível para mecanismos Valkey ou Redis OSS.

Com a Descoberta automática, seu aplicativo não precisa se conectar manualmente a nós de cache individuais. Em vez disso, seu aplicativo se conecta a um nó Memcached e recupera a lista de nós.

Nessa lista, seu aplicativo está ciente do restante dos nós no cluster e pode se conectar a qualquer um deles. Você não precisa embutir em código os endpoints de nó de cache individuais no seu aplicativo.

Se você estiver usando o tipo de rede dual stack em seu cluster, o Auto Discovery retornará somente IPv6 endereços IPv4 ou, dependendo de qual deles você selecionar. Para obter mais informações, consulte [Escolhendo um tipo de rede em ElastiCache](#).

Todos os nós de cache no cluster mantêm uma lista de metadados sobre todos os outros nós. Esses metadados são atualizados sempre que os nós são adicionados ou removidos do cluster.

Tópicos

- [Benefícios da descoberta automática com o Memcached](#)
- [Como a Descoberta automática funciona](#)
- [Como usar a Descoberta automática](#)
- [Conexão manual aos nós de cache do Memcached](#)
- [Adicionar a Descoberta automática à sua biblioteca de clientes do Memcached](#)
- [ElastiCache clientes com descoberta automática](#)

Benefícios da descoberta automática com o Memcached

Ao usar o Memcached, a descoberta automática oferece os seguintes benefícios:

- Quando você aumenta o número de nós em um cluster, os novos nós se registram no endpoint da configuração e em todos os outros nós. Quando você remove os nós do cluster de cache, os nós de partida cancelam o registro. Em ambos os casos, todos os outros nós no cluster são atualizados com os metadados do nó de cache mais recentes.
- As falhas de nós de cache são detectadas automaticamente. Os nós com falha são automaticamente substituídos.

Note

Até que a substituição dos nós seja concluída, o nó continuará a falhar.

- Um programa cliente só precisa se conectar ao endpoint da configuração. Depois disso, a biblioteca da Descoberta automática se conecta a todos os outros nós do cluster.
- Os programas cliente sondam o cluster uma vez por minuto (esse intervalo pode ser ajustado, se necessário). Se houver alguma alteração na configuração do cluster, como nós novos ou excluídos, o cliente receberá uma lista atualizada de metadados. Em seguida, o cliente se conecta ou desconecta desses nós conforme necessário.

A descoberta automática está ativada em todos os clusters do ElastiCache Memcached. Você não precisa reinicializar nenhum dos seus nós de cache para usar esse recurso.

Como a Descoberta automática funciona

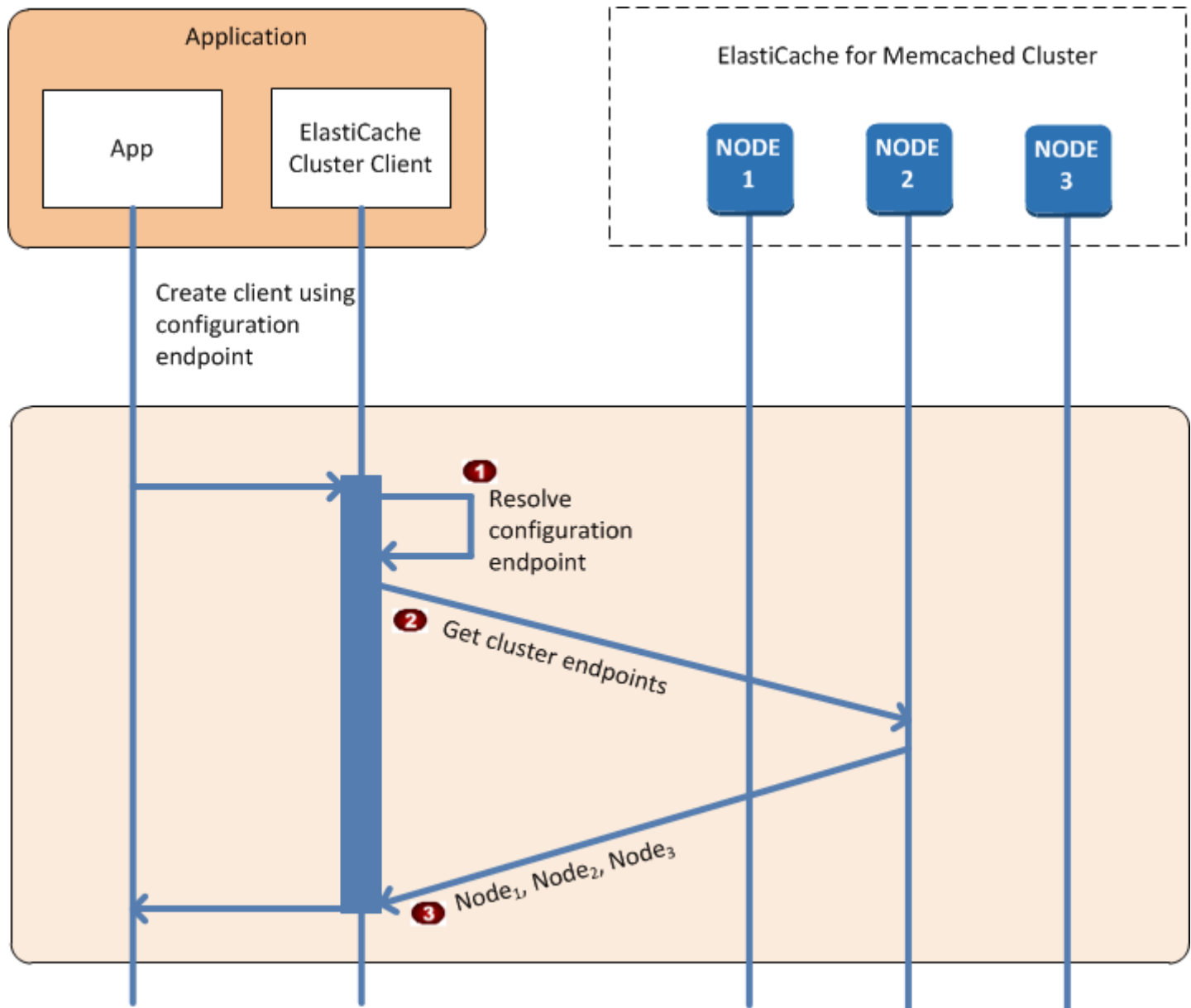
Tópicos

- [Conexão com nós de cache](#)
- [Operações normais de cluster](#)
- [Outras operações](#)

Esta seção descreve como os aplicativos cliente usam o cliente de cluster do ElastiCache para gerenciar conexões de nós de cache e interagir com itens de dados no cache.

Conexão com nós de cache

Do ponto de vista do aplicativo, conectar-se ao endpoint de configuração de cluster não é diferente de conectar-se diretamente a um nó de cache individual. O diagrama de sequência a seguir mostra o processo de conexão com nós de cache.



Processo de conexão com nós de cache

- O aplicativo resolve o nome DNS do endpoint de configuração. Como o endpoint de configuração mantém entradas CNAME para todos os nós de cache, o nome DNS é resolvido para um dos nós. O cliente pode então se conectar a esse nó.
- O cliente solicita as informações de configuração para todos os outros nós. Como cada nó mantém informações de configuração para todos os nós do cluster, qualquer nó pode transmitir informações de configuração ao cliente mediante solicitação.

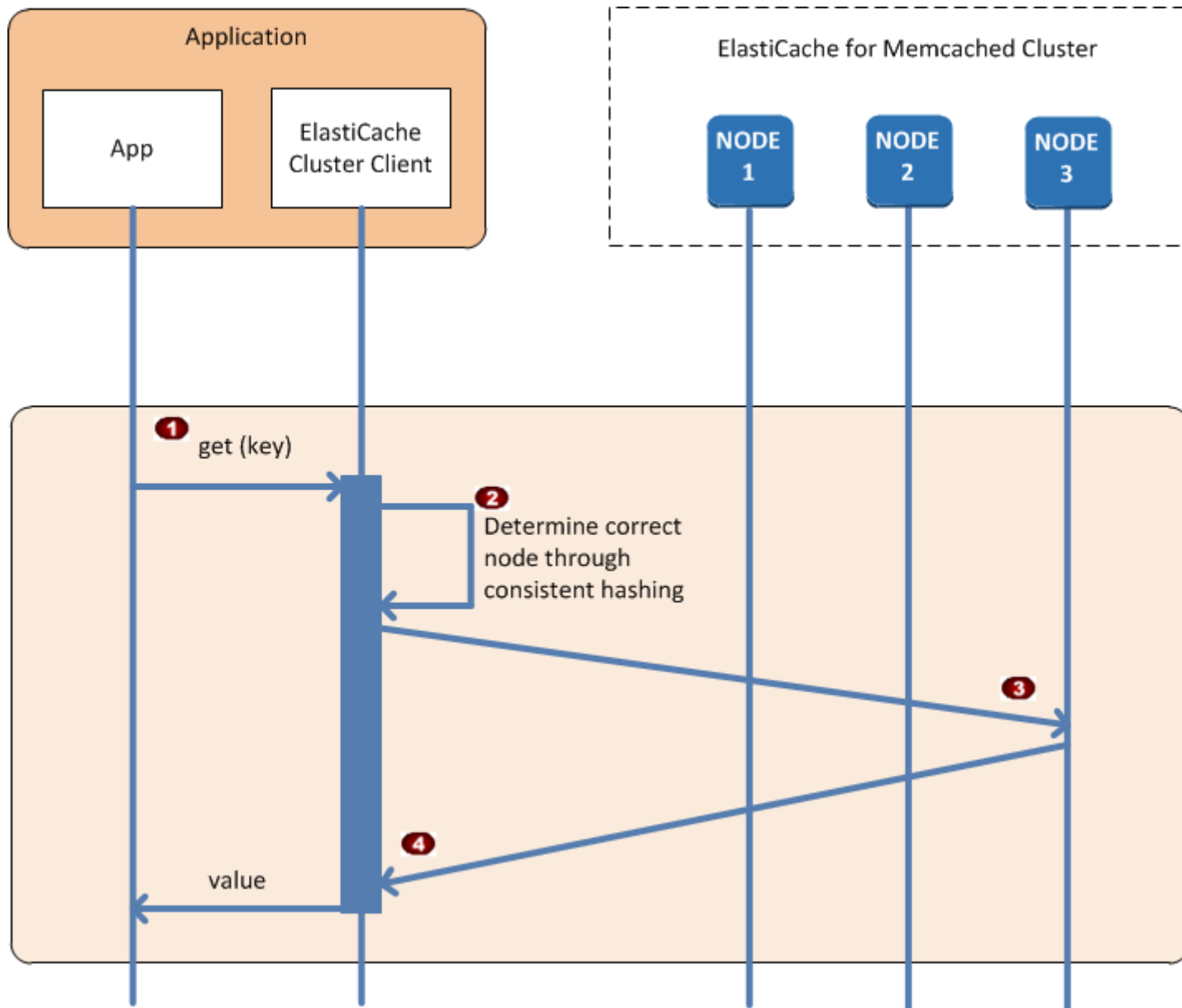
- O cliente recebe a lista atual de nomes de host e endereços IP de nós de cache. Em seguida, ele pode se conectar a todos os outros nós do cluster.

 Note

O programa cliente atualiza sua lista de nomes de host e endereços IP de nós de cache uma vez por minuto. Esse intervalo de sondagem pode ser ajustado se necessário.

Operações normais de cluster

Quando a aplicação estiver conectada a todos os nós de cache, o cliente de cluster do ElastiCache determinará quais nós deverão armazenar os itens de dados individuais e quais nós deverão ser consultados para esses itens de dados mais tarde. O diagrama de sequência a seguir mostra o processo das operações normais de cluster.



Processo das operações normais de cluster

- O aplicativo emite uma solicitação get para um item de dados específico, identificado por sua chave.
- O cliente usa um algoritmo de hash contra a chave para determinar qual nó de cache contém o item de dados.
- O item de dados é solicitado do nó apropriado.
- O item de dados é retornado ao aplicativo.

Outras operações

Em algumas situações, você pode alterar os nós de um cluster. Por exemplo, você pode acrescentar um nó adicional para acomodar demanda adicional ou excluir um nó para economizar dinheiro durante períodos de demanda reduzida. Ou você pode substituir um nó devido a uma falha de nó de uma ou outra classificação.

Quando há uma alteração no cluster que requer uma atualização de metadados para os endpoints do cluster, essa alteração é feita em todos os nós ao mesmo tempo. Portanto, os metadados em qualquer nó especificado são consistentes com os metadados em todos os outros nós do cluster.

Em cada um desses casos, os metadados são consistentes entre todos os nós em todos os momentos, já que os metadados são atualizados ao mesmo tempo para todos os nós do cluster. Você sempre deve usar o endpoint de configuração para obter os endpoints dos vários nós no cluster. Ao usar o endpoint de configuração, você garante que não obterá dados de endpoint de um nó que "desaparecerá".

Adicionar um nó

Durante o tempo em que o nó está sendo configurado, seu endpoint não é incluído nos metadados. Assim que o nó estiver disponível, ele será adicionado aos metadados de cada um dos nós do cluster. Nesse cenário, os metadados são consistentes entre todos os nós, e você poderá interagir com o novo nó somente depois que ele estiver disponível. Antes de o nó estar disponível, você não saberá sobre ele e interagirá com os nós no seu cluster como se o novo nó não existisse.

Excluir um nó

Quando um nó é removido, seu endpoint é primeiramente removido dos metadados e, em seguida, o nó é removido do cluster. Nesse cenário, os metadados em todos os nós são consistentes e em nenhuma ocasião ele conterá o endpoint para o nó a ser removido se esse nó não estiver disponível. Durante o tempo de remoção do nó, ele não é relatado nos metadados e, portanto, seu aplicativo só interagirá com os $n-1$ nós restantes, como se o nó não existisse.

Substituir um nó

Se um nó falhar, o ElastiCache o retirará e configurará uma substituição. O processo de substituição demora alguns minutos. Durante esse tempo, os metadados em todos os nós ainda mostram o endpoint do nó com falha, mas qualquer tentativa de interação com esse nó falhará. Portanto, sua lógica sempre deve incluir uma lógica de nova tentativa.

Como usar a Descoberta automática

Para começar a usar a Descoberta automática com o ElastiCache para Memcached, siga estas etapas:

- [Obter o endpoint de configuração](#)
- [Baixe o cliente de cluster do ElastiCache](#)
- [Modificar seu programa aplicativo](#)

Obter o endpoint de configuração

Para se conectar a um cluster, os programas cliente devem conhecer o endpoint de configuração de cluster. Consulte o tópico [Localização de endpoints de um cluster \(console\) \(Memcached\)](#)

Você também pode usar o comando `aws elasticache describe-cache-clusters` com o parâmetro `--show-cache-node-info`:

Seja qual for o método usado para encontrar os endpoints do cluster, o endpoint de configuração sempre terá `.cfg` em seu endereço.

Exemplo Localização de endpoints usando a AWS CLI para o ElastiCache

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws elasticache describe-cache-clusters ^
  --cache-cluster-id mycluster ^
  --show-cache-node-info
```

Esta operação produz uma saída semelhante à seguinte (formato 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"  
    }  
]  
}
```

Baixe o cliente de cluster do ElastiCache

Para tirar proveito da Descoberta automática, os programas cliente devem usar o cliente de cluster do ElastiCache. O cliente de cluster do ElastiCache está disponível para Java, PHP e .NET e contém toda a lógica necessária para descobrir e conectar-se a todos os seus nós de cache.

Para baixar o cliente de cluster do ElastiCache

1. Faça login no Console de Gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No console do ElastiCache, escolha Cliente de cluster do ElastiCache, e depois escolha Baixar.

O código-fonte do cliente de cluster do ElastiCache para Java está disponível em <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java>. Essa biblioteca se baseia no popular cliente Spymemcached. O cliente de cluster do ElastiCache é lançado sob a Licença de software da Amazon <https://aws.amazon.com/asl>. Você tem a liberdade de modificar o código-fonte como você bem entender. Você pode até mesmo incorporar o código em outras bibliotecas Memcached de código aberto ou no seu próprio código de cliente.

Note

Para usar o cliente de cluster do ElastiCache para PHP, você precisa primeiro instalá-lo na sua instância do Amazon EC2. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

Para clientes compatíveis com TLS, baixe o binário com PHP versão 7.4 ou superior.

Para usar o cliente de cluster do ElastiCache para .NET, você precisa primeiro instalá-lo na sua instância do Amazon EC2. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para .NET](#).

Modificar seu programa aplicativo

Modifique seu programa aplicativo para que ele use a Descoberta automática. As seções a seguir mostram como usar o cliente de cluster do ElastiCache para Java, PHP e .NET.

Important

Ao especificar o endpoint de configuração do cluster, certifique-se de que o endpoint tenha ".cfg" em seu endereço, conforme mostrado aqui. Não use um CNAME ou um endpoint sem ".cfg".

```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

Se o endpoint de configuração de cluster não for especificado explicitamente, a configuração ocorrerá para um nó específico.

Uso do cliente de cluster do ElastiCache para Java

O programa abaixo demonstra como usar o cliente de cluster do ElastiCache para se conectar a um endpoint de configuração de cluster e adicionar um item de dados ao cache. Usando a Descoberta automática, o programa se conecta a todos os nós do cluster sem qualquer intervenção adicional.

```
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");
}
}
```

Usando o ElastiCache Cluster Client para PHP

O programa abaixo demonstra como usar o ElastiCache Cluster Client para se conectar a um endpoint de configuração de cluster e adicionar um item de dados ao cache. Usando a Descoberta automática, o programa se conectará a todos os nós do cluster sem qualquer intervenção adicional.

Para usar o ElastiCache Cluster Client para PHP, primeiro você precisará instalá-lo na sua EC2 instância da Amazon. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

```
<?php

/**
 * Sample PHP code to show how to integrate with the Amazon ElastiCache
 * Auto Discovery feature.
 */

/* Configuration endpoint to use to initialize memcached client.
 * This is only an example. */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the ElastiCache cluster.
 * This is only an example */
$server_port = 11211;

/**
 * The following will initialize a Memcached client to utilize the Auto Discovery
 * feature.
 *
 * By configuring the client with the Dynamic client mode with single endpoint, the
 * client will periodically use the configuration endpoint to retrieve the current
 * cache
 * cluster configuration. This allows scaling the 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');
?>
```

Para ver um exemplo de como usar o ElastiCache Cluster Client com o TLS ativado, consulte [Usando criptografia em trânsito com PHP e Memcached](#).

Uso do cliente de cluster do ElastiCache para .NET

Note

O cliente de cluster ElastiCache .NET foi suspenso em maio de 2022.

O cliente .NET para o ElastiCache é um código aberto que pode ser encontrado em <https://github.com/awslabs/elasticache-cluster-config-net>.

Em geral, os aplicativos .NET obtêm suas configurações de seu arquivo de configuração. O seguinte é um exemplo de arquivo de configuração de aplicativo.

```
<?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>
```

O programa em C# abaixo demonstra como usar o cliente de cluster do ElastiCache para se conectar a um endpoint de configuração de cluster e adicionar um item de dados ao cache. Usando a Descoberta automática, o programa se conectará a todos os nós do cluster sem qualquer intervenção adicional.

```
// *****
// 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
```

Conexão manual aos nós de cache do Memcached

Se o seu programa cliente não usar a descoberta automática, ele poderá se conectar manualmente a cada um dos nós de cache do Memcached. Este é o comportamento padrão para clientes Memcached.

Você pode obter uma lista de nomes de host de nós de cache e números de portas no [Console de Gerenciamento da AWS](#). Você também pode usar o AWS CLI `aws elasticache describe-cache-clusters` comando com o `--show-cache-node-info` parâmetro.

Example

O seguinte trecho de código Java mostra como se conectar a todos os nós em um cluster de quatro nós:

```
...

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

Se você expandir ou reduzir seu cluster adicionando ou removendo nós, precisará atualizar a lista de nós no código do cliente.

Adicionar a Descoberta automática à sua biblioteca de clientes do Memcached

As informações de configuração para Descoberta automática são armazenadas de forma redundante em cada nó de cluster Memcached. Os aplicativos cliente podem consultar qualquer nó de cache e obter as informações de configuração para todos os nós no cluster.

A maneira como um aplicativo faz isso depende da versão do mecanismo de cache:

- Se a versão do mecanismo de cache for 1.4.14 ou superior, use o comando `config`.
- Se a versão do mecanismo de cache for inferior a 1.4.14, use o comando `get AmazonElastiCache:cluster`.

As saídas desses dois comandos são idênticas e estão descritas na seção [Output Format](#) abaixo.

Versão do mecanismo de cache 1.4.14 ou superior

Para o mecanismo de cache versão 1.4.14 ou superior, use o comando `config`. Esse comando foi adicionado ao Memcached ASCII e aos protocolos binários por ElastiCache, e é implementado no Cluster Client. ElastiCache Se você quiser usar a Descoberta automática com outra biblioteca de cliente, essa biblioteca precisará ser estendida para oferecer suporte ao comando `config`.

Note

A seguinte documentação pertence ao protocolo ASCII. No entanto, o comando `config` oferece suporte para ASCII e binário. Se você quiser adicionar suporte à descoberta automática usando o protocolo binário, consulte o [código-fonte do ElastiCache Cluster Client](#).

Sintaxe

```
config [sub-command] [key]
```

Opções

Name (Nome)	Description	Obrigatório
sub- command		Sim

Name (Nome)	Description	Obrigatório
	O subcomando usado para interagir com um nó de cache. Para a Descoberta automática, esse subcomando é <code>get</code> .	
<code>key</code>	A chave na qual a configuração de cluster está armazenada. Para a Descoberta automática, essa chave se chama <code>cluster</code> .	Sim

Para obter as informações de configuração do cluster, use o seguinte comando:

```
config get cluster
```

Versão do mecanismo de cache 1.4.14 ou inferior

Para obter as informações de configuração do cluster, use o seguinte comando:

```
get AmazonElastiCache:cluster
```

Note

Não altere a chave “:clusterAmazonElastiCache”, pois é aqui que residem as informações de configuração do cluster. Se você sobrescrever essa chave, o cliente poderá ser configurado incorretamente por um breve período de tempo (não mais que 15 segundos) antes de atualizar ElastiCache automaticamente e corretamente as informações de configuração.

Output Format

Se você usar `config get cluster` ou `get AmazonElastiCache:cluster`, a resposta consistirá em duas linhas:

- O número de versão das informações de configuração. Cada vez que um nó é adicionado ou removido do cluster, o número da versão aumenta em um.
- Uma lista de nós de cache. Cada nó na lista é representado por um grupo de nome do host| endereço IP|porta, e cada nó é delimitado por um espaço.

Um retorno de carro e um caractere de avanço de linha (CR + LF) aparecem no final de cada linha. A linha de dados contém um caractere de avanço de linha (LF) no final, ao qual o CR+LF é adicionado. A linha de versão de configuração é encerrada por LF sem o CR.

Um cluster contendo três nós seria representado da seguinte maneira:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Cada nó é mostrado com o CNAME e o endereço IP privado. O CNAME sempre estará presente. Se o endereço IP privado não estiver disponível, ele não será mostrado. Entretanto, os caracteres de barra vertical "|" continuarão a ser impressos.

Example

Veja a seguir um exemplo da carga útil retornada quando você consulta as informações de configuração:

```
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

- A segunda linha indica que as informações de configuração foram modificadas doze vezes até agora.
- Na terceira linha, a lista de nós está em ordem alfabética por nome de host. Essa ordem pode estar em uma sequência diferente da que você está usando no seu aplicativo cliente.

ElastiCache clientes com descoberta automática

Os programas cliente de cluster podem identificar e se conectar automaticamente a todos os nós de cluster que executem o mecanismo Memcached.

Esta seção discute a instalação e a configuração dos clientes ElastiCache PHP e .NET para uso com descoberta automática.

Tópicos

- [Instalação e compilação de clientes de cluster](#)
- [Configuração do ElastiCache](#)

Instalação e compilação de clientes de cluster

Esta seção discute a instalação, a configuração e a compilação dos clientes de cluster de descoberta automática PHP e .NET do Amazon ElastiCache.

Tópicos

- [Instalação do cliente de cluster do ElastiCache para .NET](#)
- [Instalação do cliente de cluster do ElastiCache para PHP](#)
- [Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP](#)

Instalação do cliente de cluster do ElastiCache para .NET

Você pode encontrar o código do cliente de cluster .NET do ElastiCache como código aberto em <https://github.com/awslabs/elasticache-cluster-config-net>.

Esta seção descreve como instalar, atualizar e remover os componentes .NET do cliente de cluster do ElastiCache em instâncias do Amazon EC2. Para obter mais informações sobre a descoberta automática, consulte [Identificar automaticamente nós no seu cluster \(Memcached\)](#). Para obter o código de exemplo do .NET para usar o cliente, consulte [Uso do cliente de cluster do ElastiCache para .NET](#).

Tópicos

- [Instalar o .NET](#)
- [Baixar o cliente de cluster .NET do ElastiCache para ElastiCache](#)
- [Instalar conjuntos da AWS com o NuGet](#)

Instalar o .NET

Você deve ter o .NET 3.5 ou posterior instalado para usar o SDK .NET da AWS para ElastiCache. Caso não tenha o .NET 3.5 ou posterior, pode fazer download e instalar a versão mais recente em <http://www.microsoft.com/net>.

Baixar o cliente de cluster .NET do ElastiCache para ElastiCache

Para baixar o cliente de cluster .NET do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, clique em ElastiCache Cluster Client.
3. Na lista Download ElastiCache Memcached Cluster Client, selecione .NET e clique em Download.

Instalar conjuntos da AWS com o NuGet

O NuGet é um sistema de gerenciamento de pacotes para a plataforma .NET. O NuGet está ciente das dependências de assembly e instala todos os arquivos necessários automaticamente. Os assemblies instalados pelo NuGet são armazenados com a sua solução, em vez de em uma localização central, como em Program Files, e, por isso, você pode instalar versões específicas para um aplicativo sem criar problemas de compatibilidade.

Instalar o NuGet

O NuGet pode ser instalado da galeria de instalação no MSDN; consulte <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Se você estiver usando o Visual Studio 2010 ou posterior, o NuGet será instalado automaticamente.

Você pode usar o NuGet no Gerenciador de Soluções ou no Console do Gerenciador de Pacotes.

Usar o NuGet no Gerenciador de Soluções

Para usar o NuGet no Gerenciador de Soluções do Visual Studio 2010

1. No menu Ferramentas, selecione Gerenciador de Pacotes de Biblioteca.
2. Clique em Console do Gerenciador de Pacotes.

Para usar o NuGet no Solution Explorer do Visual Studio 2012 ou Visual Studio 2013

1. No menu Ferramentas, selecione Gerenciador de Pacotes NuGet.
2. Clique em Console do Gerenciador de Pacotes.

Na linha de comando, você pode instalar os assemblies usando Install-Package, conforme mostrado a seguir.

```
Install-Package Amazon.ElastiCacheCluster
```

Para ver uma página de cada pacote disponível via NuGet, como os conjuntos SDK da AWS e extensões da AWS, consulte o site do NuGet em <http://www.nuget.org>. A página de cada pacote inclui uma linha de comando de amostra para instalar o pacote usando o console e uma lista das versões anteriores desse pacote que estão disponíveis no NuGet.

Para obter mais informações sobre os comandos do Console do Gerenciador de Pacotes, consulte <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>.

Instalação do cliente de cluster do ElastiCache para PHP

Esta seção descreve como instalar, atualizar e remover os componentes de PHP do cliente de cluster do ElastiCache em instâncias do Amazon EC2. Para obter mais informações sobre a descoberta automática, consulte [Identificar automaticamente nós no seu cluster \(Memcached\)](#). Para obter o código de exemplo do PHP para usar o cliente, consulte [Usando o ElastiCache Cluster Client para PHP](#).

Tópicos

- [Para baixar o pacote de instalação](#)
- [Para usuários que já possuam a extensão php-memcached instalada](#)
- [Etapas de instalação para novos usuários](#)
- [Remoção do cliente de cluster PHP](#)

Para baixar o pacote de instalação

Para garantir que está usando a versão correta do cliente de cluster do ElastiCache para PHP, você precisa saber qual versão do PHP está instalada na sua instância do Amazon EC2. Você também precisará saber se a sua instância do Amazon EC2 está executando uma versão de 64 bits ou 32 bits do Linux.

Para determinar a versão do PHP instalada na sua instância do Amazon EC2

- No prompt de comando, execute o seguinte comando:

```
php -v
```

A versão do PHP será mostrada na saída, como neste exemplo:

```
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

Se suas versões do PHP e do Memcached forem incompatíveis, você receberá uma mensagem de erro como a seguinte:

```
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
```

Se isso acontecer, será necessário compilar o módulo do código-fonte. Para obter mais informações, consulte [Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP](#).

Para determinar a sua arquitetura de AMI do Amazon EC2 (64 bits ou 32 bits)

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. Na lista Instances (Instâncias), clique na sua instância do Amazon EC2.
3. Na guia Description, procure o campo AMI:. Uma instância de 64 bits deve ter x86_64 como parte da descrição. Para uma instância de 32 bits, procure i386 ou i686 neste campo.

Agora, você está pronto para baixar o cliente de cluster do ElastiCache.

Para baixar o cliente de cluster do ElastiCache para PHP

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No console do ElastiCache, escolha ElastiCache Cluster Client (Cliente de cluster do ElastiCache).
3. Na lista Download ElastiCache Memcached Cluster Client (Baixar cliente de cluster Memcached do ElastiCache), escolha o cliente de cluster do ElastiCache que corresponda à sua versão do PHP e arquitetura da AMI e depois escolha o botão Download (Baixar).

Para usuários que já possuam a extensão php-memcached instalada

Para atualizar a instalação do **php-memcached**

1. Remova a instalação anterior da extensão Memcached para PHP, conforme descrito pelo tópico [Remoção do cliente de cluster PHP](#).
2. Instale a nova extensão php-memcached do ElastiCache conforme descrito anteriormente em [Etapas de instalação para novos usuários](#).

Etapas de instalação para novos usuários

Tópicos

- [Instalação do PHP 7.x para novos usuários](#)
- [Instalação do PHP 5.x para novos usuários](#)

Instalação do PHP 7.x para novos usuários

Tópicos

- [Para instalar o PHP 7 em uma AMI Ubuntu Server 14.04 LTS \(64 bits e 32 bits\)](#)
- [Para instalar o PHP 7 em uma AMI Amazon Linux 201609](#)
- [Para instalar o PHP 7 em uma AMI SUSE Linux](#)

Para instalar o PHP 7 em uma AMI Ubuntu Server 14.04 LTS (64 bits e 32 bits)

1. Execute uma nova instância a partir da AMI.
2. Execute os seguintes comandos:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Instale o PHP 7.

```
sudo yum install php70
```

4. Baixe o cliente de cluster do Amazon ElastiCache.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Extraia latest-64bit.

```
tar -zxvf latest-64bit
```

6. Com permissões de root, copie o arquivo de artefato extraído amazon-elasticache-cluster-client.so para /usr/lib/php/20151012.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. Insira a linha extension=amazon-elasticache-cluster-client.so no arquivo /etc/php/7.0/cli/php.ini.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php/7.0/cli/php.ini
```

8. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 7 em uma AMI Amazon Linux 201609

1. Execute uma nova instância a partir da AMI.
2. Execute o seguinte comando:

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

3. Instale o PHP 7.

```
sudo yum install php70
```

4. Baixe o cliente de cluster do Amazon ElastiCache.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```


5. Extraia latest-64bit.

```
tar -zxvf latest-64bit
```

6. Com permissão de root, copie o arquivo de artefato extraído amazon-elasticache-cluster-client.so para /usr/lib64/php/7.0/modules/.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. Crie o arquivo 50-memcached.ini.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php-7.0.d/50-memcached.ini
```

8. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 7 em uma AMI SUSE Linux

1. Execute uma nova instância a partir da AMI.
2. Execute o seguinte comando:

```
sudo zypper install gcc
```

3. Instale o PHP 7.

```
sudo yum install php70
```

4. Baixe o cliente de cluster do Amazon ElastiCache.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. Extraia latest-64bit.

```
tar -zxvf latest-64bit
```

6. Com permissão de root, copie o arquivo de artefato extraído amazon-elasticache-cluster-client.so para /usr/lib64/php7/extensions/.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Insira a linha extension=amazon-elasticache-cluster-client.so no arquivo /etc/php7/cli/php.ini.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php7/cli/php.ini
```

8. Inicie ou reinicie seu servidor Apache.

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

Instalação do PHP 5.x para novos usuários

Tópicos

- [Para instalar o PHP 5 em uma AMI Amazon Linux 2014.03 \(64 bits e 32 bits\)](#)
- [Para instalar o PHP 5 em uma AMI Red Hat Enterprise Linux 7.0 \(64 bits e 32 bits\)](#)
- [Para instalar o PHP 5 em uma AMI Ubuntu Server 14.04 LTS \(64 bits e 32 bits\)](#)
- [Para instalar o PHP 5 para a AMI SUSE Linux Enterprise Server 11 \(64 bits ou 32 bits\)](#)
- [Outras distribuições do Linux](#)

Para instalar o PHP 5 em uma AMI Amazon Linux 2014.03 (64 bits e 32 bits)

1. Inicie uma instância do Amazon Linux (64 bits ou 32 bits) e faça login nela.
2. Instalar dependências PHP:

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

3. Baixe o pacote php-memcached correto para sua instância do Amazon EC2 e versão do PHP. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o php-memcached. O URI deve ser o caminho de download para o pacote de instalação:

```
sudo pecl install <package download path>
```

Veja a seguir um exemplo de comando de instalação para o PHP 5.4, Linux de 64 bits. Nessa amostra, substitua *X.Y.Z* pelo número de versão atual:

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

 Note

Use a versão recente do artefato de instalação.

5. Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php.d` e insira "extension=amazon-elasticache-cluster-client.so" no arquivo:

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

6. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 5 em uma AMI Red Hat Enterprise Linux 7.0 (64 bits e 32 bits)

1. Execute uma instância do Red Hat Enterprise Linux (64 bits ou 32 bits) e faça login nela.
2. Instalar dependências PHP:

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

3. Baixe o pacote `php-memcached` correto para sua instância do Amazon EC2 e versão do PHP. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o `php-memcached`. O URI deve ser o caminho de download para o pacote de instalação:

```
sudo pecl install <package download path>
```

5. Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php.d` e insira `extension=amazon-elasticache-cluster-client.so` no arquivo.

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

6. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 5 em uma AMI Ubuntu Server 14.04 LTS (64 bits e 32 bits)

1. Inicie uma instância do Ubuntu Linux (64 bits ou 32 bits) e faça login nela.
2. Instalar dependências PHP:

```
sudo apt-get update  
sudo apt-get install gcc g++ php5 php-pear
```

3. Baixe o pacote `php-memcached` correto para sua instância do Amazon EC2 e versão do PHP. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o `php-memcached`. O URI deve ser o caminho de download para o pacote de instalação.

```
sudo pecl install <package download path>
```

Note

Essa etapa de instalação instala o artefato de build `amazon-elasticache-cluster-client.so` no diretório `/usr/lib/php5/20121212*`. Verifique o caminho absoluto do artefato de build, porque você precisa dele na próxima etapa.

Se o comando anterior não funcionar, você precisará extrair manualmente o artefato de cliente PHP `amazon-elasticache-cluster-client.so` do arquivo `*.tgz` obtido por download e copiá-lo para o diretório `/usr/lib/php5/20121212*`.

```
tar -xvf <package download path>
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php5/cli/conf.d` e insira `"extension=<absolute path to amazon-elasticache-cluster-client.so>"` no arquivo.

```
echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee
--append /etc/php5/cli/conf.d/memcached.ini
```

6. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 5 para a AMI SUSE Linux Enterprise Server 11 (64 bits ou 32 bits)

1. Inicie uma instância do SUSE Linux (64 bits ou 32 bits) e faça login nela.
2. Instalar dependências PHP:

```
sudo zypper install gcc php53-devel
```

3. Baixe o pacote `php-memcached` correto para sua instância do Amazon EC2 e versão do PHP. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o `php-memcached`. O URI deve ser o caminho de download para o pacote de instalação.

```
sudo pecl install <package download path>
```

5. Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php5/conf.d` e insira **`extension=amazon-elasticache-cluster-client.so`** no arquivo.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php5/conf.d/memcached.ini
```

6. Inicie ou reinicie seu servidor Apache.

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

Note

Se a Etapa 5 não funcionar para nenhuma das plataformas anteriores, verifique o caminho de instalação para `amazon-elasticache-cluster-client.so`. Além disso, especifique o caminho completo do binário da extensão. Além disso, verifique se o PHP em uso é uma versão com suporte. Oferecemos suporte às versões 5.3 a 5.5.

Outras distribuições do Linux

Em alguns sistemas, especialmente o CentOS7 e o Red Hat Enterprise Linux (RHEL) 7.1, `libsasl2.so.3` substituiu `libsasl2.so.2`. Nesses sistemas, quando você carrega o cliente de cluster ElastiCache, ele tenta e não consegue localizar e carregar `libsasl2.so.2`. Para resolver esse problema, crie um link simbólico para `libsasl2.so.3`, de forma que, quando o cliente tentar carregar `libsasl2.so.2`, ele seja redirecionado para `libsasl2.so.3`. O código a seguir cria esse link simbólico.

```
cd /usr/lib64
sudo ln libsasl2.so.3 libsasl2.so.2
```

Remoção do cliente de cluster PHP

Tópicos

- [Remover uma versão anterior do PHP 7](#)
- [Remover uma versão anterior do PHP 5](#)

Remover uma versão anterior do PHP 7

Para remover uma versão anterior do PHP 7

1. Remova o arquivo `amazon-elasticache-cluster-client.so` do diretório lib PHP apropriado, conforme indicado anteriormente nas instruções de instalação. Consulte a seção para a sua instalação em [Para usuários que já possuam a extensão php-memcached instalada](#).

2. Remova a linha `extension=amazon-elasticache-cluster-client.so` do arquivo `php.ini`.
3. Inicie ou reinicie seu servidor Apache.

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

Remover uma versão anterior do PHP 5

Para remover uma versão anterior do PHP 5

1. Remova a extensão `php-memcached`:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Remova o arquivo `memcached.ini` adicionado no diretório apropriado, conforme indicado nas etapas de instalação anteriores.

Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP

Esta seção discute como obter e compilar o código-fonte do cliente de cluster do ElastiCache para PHP.

Existem dois pacotes que você precisa extrair do GitHub e compilar; [aws-elasticache-cluster-client-libmemcached](https://github.com/awslabs/aws-elasticache-cluster-client-libmemcached) e [aws-elasticache-cluster-client-memcached-for-php](https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-php).

Tópicos

- [Compilação da biblioteca libmemcached](#)
- [Compilação do cliente de descoberta automática do Memcached do ElastiCache para PHP](#)

Compilação da biblioteca libmemcached

Para compilar a biblioteca aws-elasticache-cluster-client-libmemcached

1. Iniciar uma instância do Amazon EC2.
2. Instale as dependências da biblioteca.

- Na AMI Amazon Linux 201509

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- Na AMI Ubuntu 14.04

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

3. Extraia o repositório e compile o código.

Download and install <https://github.com/awslabs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz>

Compilação do cliente de descoberta automática do Memcached do ElastiCache para PHP

As seções a seguir descrevem como compilar o cliente de descoberta automática do Memcached do ElastiCache

Tópicos

- [Compilação do cliente do Memcached do ElastiCache para PHP 7](#)
- [Compilação do cliente do Memcached do ElastiCache para PHP 5](#)

Compilação do cliente do Memcached do ElastiCache para PHP 7

Execute o seguinte conjunto de comandos no diretório de código.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
git checkout php7
sudo yum install php70-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl
make
make install
```

Note

Você pode vincular estaticamente a biblioteca libmemcached ao binário PHP para que ele possa ser compatibilizado em várias plataformas Linux. Para fazer isso, execute o seguinte comando antes de make:

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

Compilação do cliente do Memcached do ElastiCache para PHP 5

Compile o aws-elasticache-cluster-client-memcached-for-php executando os seguintes comandos na pasta aws-elasticache-cluster-client-memcached-for-php/.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
```

```
make install
```

Configuração do ElastiCache

Um cluster do ElastiCache tem conformidade de protocolo com o Valkey, Memcached e Redis OSS. O código, aplicações e as ferramentas mais populares que você usa hoje com seu ambiente existente funcionarão perfeitamente com o serviço.

Esta seção discute considerações específicas para se conectar a nós de cache no ElastiCache.

Tópicos

- [Comandos restritos](#)
- [Localização de endpoints de nó e números de porta](#)
- [Conexão para uso da descoberta automática](#)
- [Conexão a nós em um cluster Valkey ou Redis OSS](#)
- [Nomes de DNS e IP subjacente](#)

Comandos restritos

Para oferecer uma experiência de serviço gerenciado, ElastiCache restringe o acesso a determinados comandos específicos do mecanismo de cache que exigem privilégios avançados. Para clusters executando o Redis OSS, os seguintes comandos não estão disponíveis:

- `bgrewriteaof`
- `bgsave`
- `config`
- `debug`
- `migrate`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Localização de endpoints de nó e números de porta

Para se conectar a um nó de cache, seu aplicativo precisa conhecer o endpoint e o número da porta desse nó.

Localização de endpoints de nó e números de porta (console)

Para determinar os endpoints do nó e os números de porta

1. Faça login no [Console de Gerenciamento do Amazon ElastiCache](#) e escolha o mecanismo em execução no seu cluster.

É exibida uma lista de todos os clusters que executam o mecanismo escolhido.

2. Continue abaixo para o mecanismo e a configuração que você está executando.
3. Escolha o nome do cluster de interesse.
4. Localize as colunas Port e Endpoint para o nó em que você está interessado.

Localização de endpoints de nó de cache e números de porta (AWS CLI)

Para determinar endpoints de nó de cache e números de porta, use o comando `describe-cache-clusters` com o parâmetro `--show-cache-node-info`.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Os nomes DNS e números de porta totalmente qualificados estão na seção Endpoint da saída.

Localização de endpoints de nó de cache e números de porta (API do ElastiCache)

Para determinar endpoints de nó de cache e números de porta, use a ação `DescribeCacheClusters` com o parâmetro `ShowCacheNodeInfo=true`.

Example

```
https://elasticache.us-west-2.amazonaws.com /  
?Action=DescribeCacheClusters  
&ShowCacheNodeInfo=true  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20140421T220302Z
&Version=2014-09-30
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20140421T220302Z
&X-Amz-Expires=20140421T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

Conexão para uso da descoberta automática

Se os seus aplicativos usam a Descoberta automática, você só precisa conhecer o endpoint de configuração do cluster, e não os endpoints individuais de cada nó de cache. Para obter mais informações, consulte [Identificar automaticamente nós no seu cluster \(Memcached\)](#).

Note

Neste momento, a Descoberta automática só está disponível para clusters executando o Memcached.

Conexão a nós em um cluster Valkey ou Redis OSS

Note

Neste momento, os clusters (API/CLI: grupos de replicação) que oferecem suporte para replicação e réplicas de leitura têm suporte apenas para clusters que executam o Valkey ou o Redis OSS.

Para clusters, o ElastiCache fornece interfaces de console, CLI e API para informações de conexão de nós individuais.

Para atividades somente leitura, os aplicativos podem se conectar a qualquer nó no cluster. No entanto, para a atividade de gravação, recomendamos que suas aplicações se conectem ao endpoint primário (Valkey ou Redis OSS (modo cluster desabilitado)) ou ao endpoint de configuração (Valkey ou Redis OSS (modo cluster habilitado)) do cluster em vez de se conectarem diretamente a um nó. Isso garantirá que seus aplicativos sempre possam localizar o nó correto, mesmo que você decida reconfigurar seu cluster promovendo uma réplica de leitura para a função primária.

Conexão a clusters em um grupo de replicação (console)

Para determinar endpoints e números de porta

- Consulte o tópico, [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#).

Conexão a clusters em um grupo de replicação (AWS CLI)

Para determinar endpoints de nó de cache e números de porta

Use o comando `describe-replication-groups` com o nome do seu grupo de replicação:

```
aws elasticache describe-replication-groups redis2x2
```

Esse comando deve produzir um resultado semelhante ao seguinte:

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "2 shards, 2 nodes (1 + 1 replica)",
      "NodeGroups": [
        {
          "Status": "available",
          "Slots": "0-8191",
          "NodeGroupId": "0001",
          "NodeGroupMembers": [
            {
              "PreferredAvailabilityZone": "us-west-2c",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-001"
            },
            {
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-002"
            }
          ]
        }
      ],
      {
        "Status": "available",
```

```

        "Slots": "8192-16383",
        "NodeGroupId": "0002",
        "NodeGroupMembers": [
            {
                "PreferredAvailabilityZone": "us-west-2b",
                "CacheNodeId": "0001",
                "CacheClusterId": "redis2x2-0002-001"
            },
            {
                "PreferredAvailabilityZone": "us-west-2a",
                "CacheNodeId": "0001",
                "CacheClusterId": "redis2x2-0002-002"
            }
        ]
    },
    "ConfigurationEndpoint": {
        "Port": 6379,
        "Address": "redis2x2.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "redis2x2",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "13:00-14:00",
    "MemberClusters": [
        "redis2x2-0001-001",
        "redis2x2-0001-002",
        "redis2x2-0002-001",
        "redis2x2-0002-002"
    ],
    "CacheNodeType": "cache.m3.medium",
    "PendingModifiedValues": {}
}
]
}

```

Conexão a clusters em um grupo de replicação (API do ElastiCache)

Para determinar endpoints de nó de cache e números de porta

Chame `DescribeReplicationGroups` com o seguinte parâmetro:

`ReplicationGroupId` = o nome do seu grupo de replicação.

Example

```
https://elasticache.us-west-2.amazonaws.com /  
  ?Action=DescribeCacheClusters  
  &ReplicationGroupId=repgroup01  
  &Version=2014-09-30  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20140421T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20140421T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20140421T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Nomes de DNS e IP subjacente

Os clientes mantêm uma lista de servidores contendo os endereços e as portas dos servidores que possuem os dados do cache. Ao usar o ElastiCache, a API DescribeCacheClusters (ou o utilitário de linha de comando describe-cache-clusters) retorna uma entrada DNS totalmente qualificada e um número de porta que podem ser usados para a lista de servidores.

Important

É importante que os aplicativos do cliente estejam configurados para resolver frequentemente os nomes DNS dos nós de cache quando eles tentam se conectar a um endpoint do nó de cache.

O ElastiCache garante que o nome DNS dos nós de cache permaneçam os mesmos quando os nós de cache forem recuperados em caso de falha.

A maioria das bibliotecas de cliente oferece suporte para conexões persistentes de nó de cache por padrão. Recomendamos o uso de conexões de nó de cache persistentes ao usar o ElastiCache. O armazenamento em cache de DNS no lado do cliente pode ocorrer em vários locais, incluindo bibliotecas de clientes, runtime da linguagem ou sistema operacional do cliente. Você deve revisar a configuração do seu aplicativo em cada camada para garantir que esteja resolvendo os endereços IP frequentemente para seus nós de cache.

Classificação de dados em níveis no ElastiCache

Os clusters do ElastiCache para Valkey ou Redis OSS que compõem um grupo de replicação e usam um tipo de nó da família r6gd têm seus dados classificados em níveis entre a memória e o armazenamento SSD (unidades de estado sólido) local. A classificação de dados em níveis fornece uma nova opção de performance de preço para workloads do Valkey ou Redis OSS ao utilizar unidades de estado sólido (SSDs) de menor custo em cada nó de cluster, além de armazenar dados na memória. Essa modalidade é ideal para workloads que acessam regularmente até 20% do conjunto de dados geral e para aplicações que podem tolerar latência adicional ao acessar dados em SSD.

Em clusters do ElastiCache com classificação de dados em níveis, o ElastiCache monitora o último horário de acesso de cada item armazenado. Quando a memória disponível (DRAM) é totalmente consumida, o ElastiCache usa um algoritmo Least-Recently Used (LRU – Usado menos recentemente) para mover automaticamente da memória para o SSD os itens acessados com pouca frequência. Quando os dados em SSD são acessados posteriormente, o ElastiCache os move de modo automático e assíncrono de volta para a memória antes de processar a solicitação. Se você tiver uma workload que acessa regularmente apenas um subconjunto de dados, a classificação de dados em níveis é uma maneira ideal de dimensionar sua capacidade de modo econômico.

Observe que, ao usar a classificação por níveis, as próprias chaves sempre permanecem na memória, enquanto a LRU controla a colocação de valores na memória versus disco. Em geral, recomendamos que seus tamanhos de chave sejam menores do que seus tamanhos de valor ao usar a classificação por níveis de dados.

A classificação de dados em níveis foi projetada para causar impacto mínimo na performance das workload da aplicação. Por exemplo, presumindo valores de string de 500 bytes, você pode esperar uma média de mais 300 microssegundos de latência para solicitações de dados armazenados em SSD em comparação com solicitações de dados em memória.

Com o maior tamanho de nó de classificação de dados em níveis (cache.r6gd.16xlarge), você pode armazenar até 1 petabyte em um só cluster de 500 nós (500 TB ao usar 1 réplica de leitura). A classificação de dados em níveis é compatível com todos os comandos e estruturas de dados do Valkey ou Redis OSS compatíveis com o ElastiCache. Para usar esse recurso, não é necessário promover alterações no lado do cliente.

Tópicos

- [Práticas recomendadas](#)

- [Limitações](#)
- [Preços](#)
- [Monitoramento](#)
- [Como usar a classificação de dados em níveis](#)
- [Como restaurar dados do backup para clusters com a classificação de dados em níveis ativada](#)

Práticas recomendadas

Recomendamos seguir estas práticas recomendadas:

- A classificação de dados em níveis é ideal para workloads que acessam regularmente até 20% do conjunto de dados geral e para aplicações que podem tolerar latência adicional ao acessar dados em SSD.
- Ao usar a capacidade SSD disponível em nós em níveis de dados, recomendamos que o tamanho do valor seja maior do que o tamanho da chave. Quando os itens são movidos entre DRAM e SSD, as chaves sempre permanecerão na memória e somente os valores serão movidos para a camada SSD.

Limitações

A classificação de dados em níveis tem as seguintes limitações:

- Você só pode usar a classificação de dados em níveis em clusters que fazem parte de um grupo de replicação.
- O tipo de nó usado deve ser da família r6gd, que está disponível nas seguintes regiões: us-east-2, us-east-1, us-west-2, us-west-1, eu-west-1, eu-central-1, eu-north-1, eu-west-3, ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-south-1, ca-central-1 e sa-east-1.
- Você deve usar um mecanismo que seja Valkey 7.2 ou posterior, ou um Redis OSS 6.2 ou posterior.
- Você não pode restaurar um backup de um cluster r6gd para outro cluster, a menos que ele também use r6gd.
- Você não pode exportar um backup para o Amazon S3 para clusters de classificação de dados em níveis.
- Não há compatibilidade para migração online com clusters em execução no tipo de nó r6gd.

- Não há compatibilidade com escalabilidade de um cluster de classificação de dados em níveis (p. ex., um cluster que use um tipo de nó r6gd) para um cluster sem classificação de dados em níveis (p. ex., um cluster que use um tipo de nó r6g). Para obter mais informações, consulte [Dimensionamento ElastiCache](#).
- O ajuste de escala automático é aceito em clusters que usam classificação de dados em níveis para Valkey versão 7.2 e posteriores e Redis OSS versão 7.0.7 e posteriores. Para obter mais informações, consulte [Auto Scaling de clusters Valkey e Redis OSS](#).
- A divisão de dados em camadas só são compatíveis com as políticas volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu e noeviction.
- A gravação sem bifurcação é compatível com o Valkey versão 7.2 e posteriores, e com o Redis OSS versão 7.0.7 e posteriores. Para obter mais informações, consulte [Como a sincronização e o backup são implementados](#).
- Itens maiores que 128 MiB não são movidos para o SSD.
- A partir do Valkey 8.1 e versões posteriores, um item cujo tamanho de chave+valor seja menor que 40 bytes não será movido para o SSD.

Preços

Os nós R6gd têm 4,8x mais capacidade total (memória + SSD) e podem ajudar você a obter mais de 60% de economia para execução com utilização máxima em comparação aos nós R6g (somente memória). Para mais informações, consulte [Preços do Amazon ElastiCache](#).

Monitoramento

O ElastiCache oferece métricas especificamente projetadas para monitorar os clusters de performance que usam a classificação de dados em níveis. Para monitorar a proporção de itens na DRAM em comparação com o SSD, é possível usar a métrica `CurrlItems` em [Métricas para Valkey e Redis OSS](#). É possível calcular a porcentagem como: $(\text{CurrlItems com dimensão: Nível} = \text{Memória} * 100) / (\text{CurrlItems sem filtro de dimensão})$.

Se a política de remoção configurada permitir, o ElastiCache começará a remover itens quando a porcentagem de itens na memória cair abaixo de 5%. Nos nós configurados com a política de não remoção, as operações de gravação receberão um erro de falta de memória.

Ainda é recomendável que você considere aumentar a escala horizontalmente para clusters de modo cluster habilitado ou aumentar a escala verticalmente para clusters de modo cluster desabilitado.

quando a porcentagem de itens na memória cair abaixo de 5%. Para obter mais informações sobre a escalabilidade, consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#). Para obter mais informações sobre métricas para clusters do Valkey ou Redis OSS que usam a classificação de dados em níveis, consulte [Métricas para o Valkey e Redis OSS](#).

Como usar a classificação de dados em níveis

Como usar a classificação de dados em níveis usando o Console de gerenciamento da AWS

Ao criar um cluster como parte de um grupo de replicação, você usa a classificação de dados em níveis selecionando um tipo de nó da família r6gd, p. ex., cache.r6gd.xlarge. A seleção desse tipo de nó ativa automaticamente a classificação de dados em níveis.

Para mais informações sobre como criar um cluster, consulte [Criação de um cluster do Valkey ou Redis OSS](#).

Como habilitar a classificação de dados em níveis usando a AWS CLI

Ao criar um grupo de replicação usando a AWS CLI, você usa a classificação de dados em níveis selecionando um tipo de nó da família r6gd, p. ex., cache.r6gd.xlarge e configurando o parâmetro `--data-tiering-enabled`.

Você não pode optar por não usar a classificação de dados em níveis ao selecionar um tipo de nó da família r6gd. Se você configurar o parâmetro `--no-data-tiering-enabled`, a operação falhará.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis OSS cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled
```

Para Windows:

```
aws elasticache create-replication-group ^
```

```
--replication-group-id redis-dt-cluster ^
--replication-group-description "Redis OSS cluster with data tiering" ^
--num-node-groups 1 ^
--replicas-per-node-group 1 ^
--cache-node-type cache.r6gd.xlarge ^
--engine redis ^
--cache-subnet-group-name default ^
--automatic-failover-enabled ^
--data-tiering-enabled
```

Após executar essa operação, você verá uma resposta semelhante ao seguinte:

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "redis-dt-cluster",
    "Description": "Redis OSS cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Como restaurar dados do backup para clusters com a classificação de dados em níveis ativada

Você pode usar o (Console), a (AWS CLI) ou a (API do ElastiCache) para restaurar um backup para um novo cluster com a classificação de dados em níveis ativada. Ao criar um cluster usando tipos de nós na família r6gd, a classificação de dados em níveis é ativada.

Como restaurar dados do backup para clusters com a classificação de dados em níveis ativada (console)

Para restaurar um backup para um novo cluster com a classificação de dados em níveis ativada (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup do qual você deseja restaurar.
4. Escolha Restore.
5. Preencha a caixa de diálogo Restore Cluster. Certifique-se de preencher todos os campos Obrigatórios e qualquer outro que você deseja alterar em relação aos padrões.
 1. Cluster ID (ID do cluster): obrigatório. O nome do novo cluster.
 2. Modo cluster habilitado (aumento de escala horizontalmente): escolha esta opção para um cluster do Valkey ou Redis OSS (modo cluster habilitado).
 3. Node Type (Tipo de nó) – Especifique cache.r6gd.xlarge ou qualquer outro tipo de nó da família r6gd.
 4. Number of Shards – Escolha o número de estilhaços desejados no novo cluster (API/CLI: grupos de nós).
 5. Replicas per Shard – Escolha o número de nós de réplica de leitura desejados em cada estilhaço.
 6. Slots and keyspaces (Slots e espaços de chaves): escolha como deseja que as chaves sejam distribuídas entre os fragmentos. Se você optar por especificar as distribuições de chaves, complete a tabela especificando os intervalos de chaves para cada estilhaço.
 7. Availability zone(s) – especifique como você deseja que as zonas de disponibilidade do cluster sejam selecionadas.
 8. Port – Somente altere esse valor se quiser que o novo cluster use uma porta diferente.
 9. Choose a VPC – Escolha a VPC na qual criar esse cluster.
 10. Parameter Group: escolha um grupo de parâmetros que reserve memória suficiente para a sobrecarga do Valkey ou Redis OSS para o tipo de nó selecionado.
6. Quando estiver satisfeito com as configurações, escolha Create (Criar).

Para mais informações sobre como criar um cluster, consulte [Criação de um cluster do Valkey ou Redis OSS](#).

Como restaurar dados do backup para clusters com a classificação de dados em níveis ativada (AWS CLI)

Ao criar um grupo de replicação usando a AWS CLI, a classificação de dados em níveis é utilizada por padrão mediante a seleção de um tipo de nó da família r6gd, p. ex., cache.r6gd.xlarge e a configuração do parâmetro `--data-tiering-enabled`.

Você não pode optar por não usar a classificação de dados em níveis ao selecionar um tipo de nó da família r6gd. Se você configurar o parâmetro `--no-data-tiering-enabled`, a operação falhará.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis OSS cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled \  
  --snapshot-name my-snapshot
```

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group ^  
  --replication-group-id redis-dt-cluster ^  
  --replication-group-description "Redis OSS cluster with data tiering" ^  
  --num-node-groups 1 ^  
  --replicas-per-node-group 1 ^  
  --cache-node-type cache.r6gd.xlarge ^  
  --engine redis ^  
  --cache-subnet-group-name default ^  
  --automatic-failover-enabled ^  
  --data-tiering-enabled ^  
  --snapshot-name my-snapshot
```

Após executar essa operação, você verá uma resposta semelhante ao seguinte:

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "redis-dt-cluster",
    "Description": "Redis OSS cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Preparação de um cluster no ElastiCache

Veja a seguir instruções sobre como criar um cluster usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Você também pode criar um cluster do ElastiCache usando [AWS CloudFormation](#). Para obter mais informações, consulte [::AWS::ElastiCache::CacheCluster](#) no AWSGuia do usuário do CloudFormation, que inclui orientações sobre como implementar essa abordagem.

Sempre que você criar um cluster ou grupo de replicação, é uma boa ideia fazer algum trabalho preparatório para que você não precise atualizar nem fazer alterações imediatamente.

Tópicos

- [Determinar os requisitos do seu cluster do ElastiCache](#)
- [Escolha do tamanho do nó](#)

Determinar os requisitos do seu cluster do ElastiCache

Preparação

Conhecer as respostas às seguintes perguntas ajuda a simplificar a criação de um cluster do ElastiCache:

- De qual tipo de instância de nó você precisa?

Para obter orientação sobre como escolher um tipo de nó de instância, consulte [Escolha do tamanho do nó](#).

- Você executará seu cluster em uma nuvem privada virtual (VPC) baseada na Amazon VPC?

Important

Se você estiver executando seu cluster em uma VPC, crie um grupo de sub-redes na mesma VPC antes de começar a criar um cluster. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).

O ElastiCache foi projetado para ser acessado de dentro da AWS usando o Amazon EC2. No entanto, ao iniciá-lo em uma VPC com base na Amazon VPC e o seu cluster em uma VPC, você pode fornecer acesso de fora da AWS. Para obter mais informações, consulte [Acessando ElastiCache recursos de fora AWS](#).

- Você precisa personalizar qualquer valor de parâmetro?

Se você fizer isso, crie um grupo de parâmetro personalizado. Para obter mais informações, consulte [Criação de um grupo de parâmetros do ElastiCache](#).

Se você estiver executando o Valkey ou o Redis OSS, considere definir `reserved-memory` ou `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

- Você precisa criar o próprio grupo de segurança da VPC?

Para obter mais informações, consulte [Segurança na sua VPC](#).

- Como você pretende implementar a tolerância a falhas?

Para obter mais informações, consulte [Atenuar falhas](#).

Tópicos

- [Requisitos de memória e processador do ElastiCache](#)
- [Configuração do cluster do Memcached](#)

- [Configuração do cluster do Valkey e Redis OSS](#)
- [Requisitos de escalabilidade do ElastiCache](#)
- [Requisitos de acesso do ElastiCache](#)
- [Requisitos de região, zona de disponibilidade e zona local do ElastiCache](#)

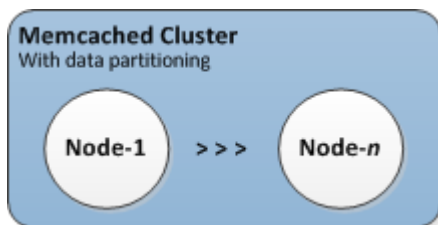
Requisitos de memória e processador do ElastiCache

O bloco de construção básico do Amazon ElastiCache é o nó. Nós são configurados de forma singular ou em agrupamentos para formar clusters. Ao determinar o tipo de nó a ser usado para o seu cluster, considere a configuração do nó do cluster e a quantidade de dados que você deve armazenar.

O mecanismo Memcached é multi-threaded e, portanto, o número de núcleos de um nó afeta o poder computacional disponível para o cluster.

Configuração do cluster do Memcached

Os clusters do ElastiCache for Memcached são compostos de 1 a 60 nós. Os dados em um cluster Memcached são particionados nos nós do cluster. Seu aplicativo conecta-se a um cluster Memcached usando um endereço de rede chamado de Endpoint. Cada nó em um cluster Memcached tem seu próprio endpoint, que seu aplicativo usa para ler ou gravar de/em um nó específico. Além dos pontos de extremidade do nó, o cluster do Memcached em si tem um endpoint chamado endpoint de configuração. Sua aplicação pode usar esse endpoint para ler ou gravar no cluster, deixando a determinação de qual nó deve ser lido ou gravado para até .

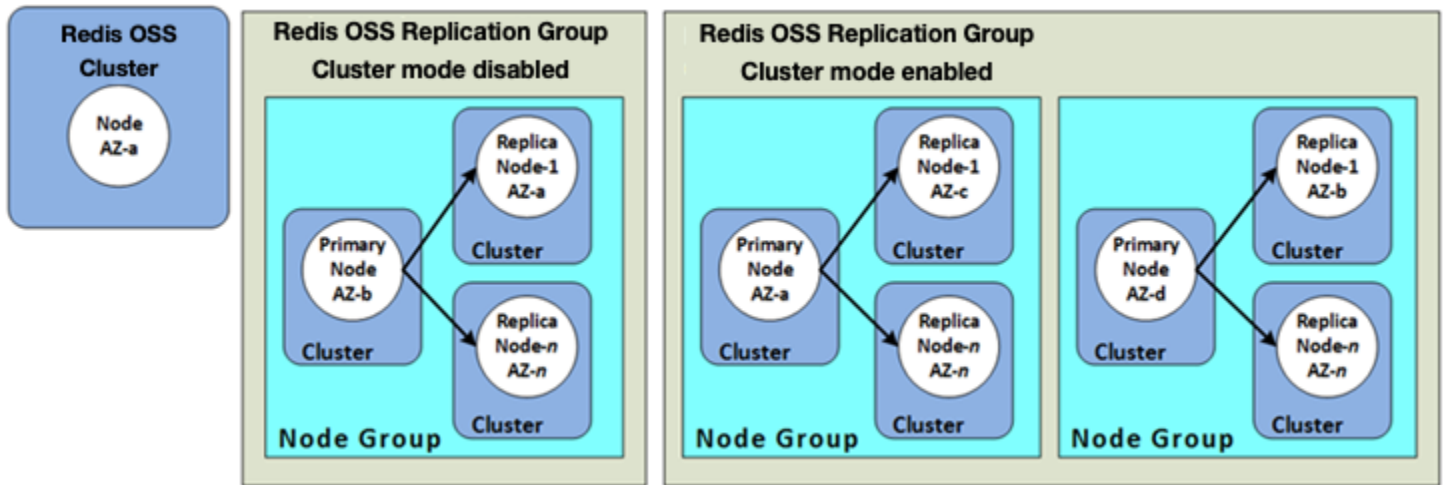


Para obter mais informações, consulte [Gerenciamento de clusters no ElastiCache](#).

Configuração do cluster do Valkey e Redis OSS

Os clusters do ElastiCache para Valkey e Redis OSS são compostos de 0 a 500 fragmentos (também chamados de grupos de nós). Os dados em um cluster do Valkey ou Redis OSS são particionados nos fragmentos do cluster. Sua aplicação se conecta a um cluster do Valkey ou Redis OSS usando um endereço de rede chamado de Endpoint. Os nós em um fragmento do Valkey ou Redis OSS cumprem uma das duas funções: uma primária de leitura/gravação e todos os outros nós

secundários somente leitura (também chamados de réplicas de leitura). Além dos endpoints do nó, o cluster do Valkey ou Redis OSS em si tem um endpoint chamado endpoint de configuração. Sua aplicação pode usar esse endpoint para ler ou gravar no cluster, deixando a determinação de qual nó deve ser lido ou gravado para até ElastiCache para Redis OSS.



Para obter mais informações, consulte [Gerenciamento de clusters no ElastiCache](#).

Requisitos de escalabilidade do ElastiCache

Todos os clusters podem ser expandidos criando um novo cluster com o novo tipo de nó maior. Quando você aumenta a escala na vertical de um cluster do Memcached, o novo cluster começa vazio. Ao aumentar a escala na vertical de um cluster do Valkey ou Redis OSS, você pode propagá-lo de um backup e evitar que o novo cluster seja iniciado vazio.

Os clusters do Amazon ElastiCache for Memcached podem ter aumento ou redução de escala na horizontal. Para expandir ou reduzir um cluster Memcached, basta adicionar ou remover nós do cluster. Se você tiver habilitado a Descoberta automática e seu aplicativo estiver se conectando ao endpoint de configuração do cluster, não será necessário fazer alterações no aplicativo quando você adicionar ou remover nós.

Para obter mais informações, consulte [Dimensionamento ElastiCache](#) neste guia.

Requisitos de acesso do ElastiCache

Por design, os clusters do Amazon ElastiCache são acessados a partir de instâncias do Amazon EC2. O acesso via rede a um cluster do ElastiCache é limitado à conta que criou esse cluster. Portanto, antes de poder acessar um cluster de uma instância do Amazon EC2, você deve autorizar a instância do Amazon EC2 a acessar o cluster. As etapas para essa autorização variam, dependendo de você ter executado na EC2-VPC ou no EC2-Classic.

Se você tiver executado seu cluster na EC2-VPC, deverá conceder entrada de rede ao cluster. Se você executou seu cluster no EC2-Classic, será necessário conceder acesso ao grupo de segurança do Amazon Elastic Compute Cloud para o grupo de segurança do ElastiCache associado à instância. Para obter instruções detalhadas, consulte [Etapa 3. Autorizar o acesso ao cluster](#) neste guia.

Requisitos de região, zona de disponibilidade e zona local do ElastiCache

O Amazon ElastiCache oferece suporte a todas as regiões da AWS. Ao colocar seus clusters do ElastiCache em uma região da AWS próxima à sua aplicação, é possível reduzir a latência. Se o seu cluster tiver vários nós, a localização deles em diferentes zonas de disponibilidade ou em zonas locais poderá reduzir o impacto das falhas no cluster.

Para obter mais informações, consulte:

- [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#)
- [Uso de zonas locais com o ElastiCache](#)
- [Atenuar falhas](#)

Escolha do tamanho do nó

O tamanho do nó selecionado para seu cluster do ElastiCache afeta os custos, o desempenho e a tolerância a falhas.

Tamanho do nó (Valkey e Redis OSS)

Para receber informações sobre os benefícios dos processadores Graviton, consulte [Processador Graviton da AWS](#).

Responder às seguintes perguntas poderá ajudá-lo a determinar o tipo mínimo de nó de que você precisa para sua implementação do Valkey ou Redis OSS:

- Você espera workloads limitadas ao throughput com várias conexões de clientes?

Se for esse o caso e você estiver executando o Redis OSS versão 5.0.6 ou posterior, você poderá obter melhor throughput e latência usando nosso atributo de E/S aprimorada, em que as CPUs disponíveis são usadas para descarregar as conexões do cliente, em nome do mecanismo Redis OSS. Se você estiver executando o Redis OSS versão 7.0.4 ou posterior, além da E/S aprimorada, você obterá aceleração adicional com multiplexação de E/S aprimorada, em que cada thread de E/S de rede dedicado canaliza comandos de vários clientes para o mecanismo Redis OSS, aproveitando a capacidade do Redis OSS de processar comandos em lotes com eficiência. No

ElastiCache para Redis OSS v7.1 e versões posteriores, estendemos a funcionalidade de threads de E/S melhorada para também lidar com a lógica da camada de apresentação. Por camada de apresentação, o que queremos dizer é que os threads de E/S aprimorada agora não estão apenas lendo a entrada do cliente, mas também analisando a entrada no formato de comando binário do Redis OSS, que acaba sendo encaminhado para o thread principal para execução, proporcionando ganho de desempenho. Consulte a [publicação do blog](#) e a página de [versões compatíveis](#) para obter detalhes adicionais.

- Você tem workloads que acessam regularmente um pequeno percentual de seus dados?

Caso a resposta seja afirmativa e seu sistema esteja em execução no mecanismo Redis OSS versão 6.2 ou posterior, você pode aproveitar a classificação de dados em níveis escolhendo o tipo de nó r6gd. Com a classificação de dados em níveis, os dados usados menos recentemente são armazenados em SSD. Quando eles são recuperados, há um pequeno custo de latência, que é equilibrado pela economia de custos. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- Quanto de memória total você precisa para seus dados?

Para obter uma estimativa geral, tome o tamanho dos itens que você deseja armazenar em cache. Multiplique esse tamanho pelo número de itens que deseja manter no cache ao mesmo tempo. Para obter uma estimativa razoável do tamanho dos itens, primeiro serialize seus itens de cache, e depois conte os caracteres. Em seguida, divida isso pelo número de fragmentos no cluster.

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- Qual versão do Redis OSS você está executando?

Versões do Redis OSS anteriores à 2.8.22 exigem que você reserve mais memória para failover, snapshot, sincronização e promoção de uma réplica para operações primárias. Esse requisito é importante porque você deve ter memória suficiente disponível para todas as gravações que ocorrem durante o processo.

O Redis OSS versão 2.8.22 e posteriores usam um processo de salvamento sem bifurcação que requer menos memória disponível que o processo anterior.

Para obter mais informações, consulte:

- [Como a sincronização e o backup são implementados](#)
- [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#)

- Qual é a intensidade de gravação do seu aplicativo?

Gravar aplicativos pesados pode exigir significativamente mais memória disponível, memória não utilizada por dados, ao tirar snapshots ou fazer failover. Sempre que o processo BGSAVE for executado, você deve ter memória suficiente que não é usada pelos dados para acomodar todas as gravações que transpirem durante o processo BGSAVE. Exemplos são ao tirar um snapshot, ao sincronizar um cluster primário com uma réplica em um cluster e ao ativar o recurso de arquivo somente anexado (AOF). Outro exemplo acontece ao promover uma réplica para primária (se você tiver o Multi-AZ habilitado). O pior caso é quando todos os seus dados são reescritos durante o processo. Nesse caso, você precisa de um tamanho de instância de nó com o dobro da memória que é necessária para os dados isoladamente.

Para obter mais informações detalhadas, consulte [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#).

- Sua implementação será um cluster autônomo do Valkey ou Redis OSS (modo cluster desabilitado) ou um cluster do Valkey ou Redis OSS (modo cluster habilitado) com vários fragmentos?

Cluster do Valkey ou Redis OSS (modo cluster desabilitado)

Se você estiver implementando um cluster do Valkey ou Redis OSS (modo cluster desabilitado), seu tipo de nó deve ser capaz de acomodar todos os seus dados mais a sobrecarga necessária, conforme descrito anteriormente.

Por exemplo, suponha que você estime que o tamanho total de todos os seus itens é de 12 GB. Nesse caso, você pode usar um nó cache.m3.xlarge com 13,3 GB de memória ou um nó cache.r3.large com 13,5 GB de memória. No entanto, talvez você precise de mais memória para operações BGSAVE. Se a sua aplicação for de escrita pesada, duplique os requisitos de memória para pelo menos 24 GB. Assim, use ou cache.m3.2xlarge com 27,9 GB de memória ou um cache.r3.xlarge com 30,5 GB de memória.

Valkey ou Redis OSS (modo cluster habilitado) com vários fragmentos

Se você estiver implementando um cluster do Valkey ou Redis OSS (modo cluster habilitado) com vários fragmentos, o tipo de nó deve ser capaz de acomodar `bytes-for-data-and-overhead / number-of-shards` bytes de dados.

Por exemplo, suponha que você estime que o tamanho total de todos os seus itens é de 12 GB e você tem dois fragmentos. Nesse caso, você pode usar um nó cache.m3.large com 6,05

GB de memória (12 GB/2). No entanto, talvez você precise de mais memória para operações BGSAVE. Se a sua aplicação for de escrita pesada, duplique os requisitos de memória para pelo menos 12 GB por fragmento. Assim, use ou `cache.m3.xlarge` com 13,3 GB de memória ou um `cache.r3.large` com 13,5 GB de memória.

- Você está usando Local Zones?

[Local Zones](#) (Zonas locais) permitem que você coloque recursos como um cluster do ElastiCache em vários locais próximos aos usuários. Mas, ao escolher o tamanho do nó, esteja ciente de que os tamanhos de nó disponíveis estão limitados ao seguinte no momento, independentemente dos requisitos de capacidade:

- Geração atual:

Tipos de nó M: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

Tipos de nó R: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

Tipos de nó T: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Enquanto seu cluster está em execução, você pode monitorar o uso da memória, a utilização do processador, o cache e as métricas de perdas no cache que são publicadas no CloudWatch. Você pode notar que seu cluster não tem a taxa de acertos desejada ou que as chaves estão sendo removidas com muita frequência. Nesses casos, você pode escolher um tamanho de nó diferente com especificações de CPU e memória maiores.

Ao monitorar o uso da CPU, lembre-se que o Valkey e o Redis OSS são de thread único. Assim, multiplique o uso da CPU relatado pelo número de núcleos da CPU para obter esse uso real. Por exemplo, uma CPU de quatro núcleos que informa uma taxa de uso de 20% é, na verdade, o único núcleo que o Redis OSS está executando com 80% de utilização.

Tamanho do nó (Memcached)

Os clusters Memcached contêm um ou mais nós com os dados do cluster particionados entre os nós. Por isso, as necessidades de memória do cluster e de memória de um nó estão relacionadas, mas não são idênticas. Você pode obter a capacidade de memória de cluster necessária tendo alguns nós grandes ou vários nós menores. Além disso, conforme suas necessidades mudarem, você poderá adicionar ou remover nós do cluster e, assim, pagar apenas pelo que precisa.

A capacidade de memória total do cluster é calculada multiplicando o número de nós no cluster pela capacidade de RAM de cada nó, depois de deduzir as despesas gerais do sistema. A capacidade de cada nó é baseada no tipo de nó.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

O número de nós no cluster é um fator chave na disponibilidade do seu cluster executando o Memcached. A falha de um único nó pode ter um impacto na disponibilidade da sua aplicação e na carga do seu banco de dados de backend. Nesse caso, o ElastiCache provisiona uma substituição para um nó com falha e ele é preenchido novamente. Para reduzir esse impacto na disponibilidade, espalhe sua memória e capacidade de computação ao redor de um número maior de nós com menor capacidade, em vez de usar um número menor de nós de alta capacidade.

Em um cenário em que você deseja ter 35 GB de memória cache, você pode definir qualquer uma das seguintes configurações:

- 11cache.t2.medium nós com 3,22 GB de memória e 2 threads cada = 35,42 GB e 22 threads.
- 6cache.m4.large nós com 6,42 GB de memória e 2 threads cada = 38,52 GB e 12 threads.
- 3cache.r4.large nós com 12,3 GB de memória e 2 threads cada = 36,90 GB e 6 threads.
- 3cache.m4.xlarge nós com 14,28 GB de memória e 4 threads cada = 42,84 GB e 12 threads.

Comparar opções de nós

Tipo de nó	Memória (em GB)	Núcleos	Custo horário*	Nós necessários	Memória total (em GB)	Total de núcleos	Custo mensal
cache.t2.medium	3.22	2	0,068 USD	11	35.42	22	US\$ 538,56
cache.m4.large	6.42	2	0,156 USD	6	38.52	12	US\$ 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

Tipo de nó	Memória (em GB)	Núcleos	Custo horário*	Nós necessários	Memória total (em GB)	Total de núcleos	Custo mensal
cache.m6g.large	6.85	2	US\$ 0,147	6	41.1	12	US\$ 635
cache.r4.large	12.3	2	0,228 USD	3	36.9	6	492,48 USD
cache.r5.large	13.07	2	US\$ 0,216	3	39.22	6	US\$ 466,56
cache.r6g.large	13.07	2	US\$ 0,205	3	42.12	6	US\$ 442

* Custo horário por nó em 8 de outubro de 2020.

Custo mensal a 100% de uso por 30 dias (720 horas).

Essas opções oferecem uma capacidade de memória semelhante, mas uma capacidade e custo computacional diferentes. Para comparar os custos das suas opções específicas, consulte [Definição de preço do Amazon ElastiCache](#).

Para clusters executados no Memcached, algumas das memórias disponíveis em cada nó são usadas para sobrecarga de conexão. Para obter mais informações, consulte [Sobrecarga de conexões do Memcached](#)

O uso de vários nós exigirá a distribuição das chaves entre eles. Cada nó possui seu próprio endpoint. Para o fácil gerenciamento de endpoints, você pode usar o recurso de descoberta automática do ElastiCache, que permite que os programas clientes identifiquem automaticamente todos os nós em um cluster. Para obter mais informações, consulte [Identificar automaticamente nós no seu cluster \(Memcached\)](#).

Em alguns casos, você pode não ter certeza de quanta capacidade precisa. Em caso afirmativo, para testes recomendamos começar com um nó `cache.m5.large`. Em seguida, monitore o uso da memória, a utilização da CPU e a taxa de acerto do cache com as métricas do ElastiCache publicadas no Amazon CloudWatch. Para obter mais informações sobre métricas do CloudWatch

para o ElastiCache, consulte [Monitorando o uso com CloudWatch métricas](#). Para produção e maiores workloads, os nós R5 fornecem o melhor desempenho e valor de custo de RAM.

Se o seu cluster não tiver a taxa de acerto desejada, você poderá adicionar facilmente mais nós, aumentando assim a memória total disponível no seu cluster.

Se o seu cluster for limitado por CPU, mas tiver taxa de acerto suficiente, tente configurar um novo cluster com um tipo de nó que forneça mais poder computacional.

Criação de um cluster do Valkey ou Redis OSS

Os exemplos a seguir mostram como criar um cluster OSS Valkey ou Redis usando a Console de gerenciamento da AWS API e AWS CLI ElastiCache

Criação de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) (console)

ElastiCache oferece suporte à replicação quando você usa o mecanismo Valkey ou Redis OSS. Para monitorar a latência entre o momento em que os dados são gravados em um cluster read/write primário Valkey ou Redis OSS e o momento em que são propagados para um cluster secundário somente para leitura, ElastiCache adiciona ao cluster uma chave especial, `ElastiCacheMasterReplicationTimestamp`. Essa chave é a hora em Tempo Universal Coordenado (UTC) atual. Como um cluster do Valkey ou Redis OSS pode ser adicionado a um grupo de replicação mais tarde, essa chave está incluída em todos os clusters do Valkey ou Redis OSS, mesmo que inicialmente eles não sejam membros de um grupo de replicação. Para obter mais informações sobre grupos de replicação, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

Para criar um cluster do Valkey ou Redis OSS (modo de cluster desabilitado), siga as etapas em [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#).

Assim que o status do seu cluster estiver disponível, você poderá conceder EC2 à Amazon acesso a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4: Conectar-se ao nó de um cluster](#).

Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Criação de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (console)

Se você estiver executando o Redis OSS 3.2.4 ou posterior, será possível criar um cluster do Valkey ou Redis OSS (modo cluster habilitado). Clusters do Valkey ou Redis OSS (modo cluster habilitado) oferecem suporte para o particionamento dos seus dados entre 1 e 500 fragmentos (API/CLI: grupos de nós), mas com algumas limitações. Para obter uma comparação entre o Valkey ou Redis OSS

(modo cluster desabilitado) e o Valkey ou Redis OSS (modo cluster habilitado), consulte [Mecanismos e versões compatíveis](#).

Para criar um cluster Valkey ou Redis OSS (modo de cluster ativado) usando o console ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região na qual você deseja iniciar esse cluster.
3. No painel de navegação, escolha Get started (Conceitos básicos).
4. Escolha Create VPC (Criar VPC) e siga as etapas em [Creating a Virtual Private Cloud \(VPC\)](#) (Criação de uma nuvem privada virtual (VPC)).
5. Na página do ElastiCache painel, escolha Criar cluster e, em seguida, escolha Criar cluster Valkey ou Criar cluster Redis OSS.
6. Em Cluster settings (Configurações do cluster), faça o seguinte:
 - a. Selecione Configure and create a new cluster (Configurar e criar um novo cluster).
 - b. Para o Cluster mode Modo cluster, escolha Enabled (Habilitado).
 - c. Para o Cluster info (Informações sobre o cluster), insira um valor para Name (Nome).
 - d. (Opcional) Insira um valor para Description (Descrição).
7. Em Location (Local):

AWS Cloud

1. Para AWS Cloud (Nuvem), recomendamos que você aceite as configurações padrão de Multi-AZ e Failover automático. Para obter mais informações, consulte [Minimizando o tempo de inatividade no ElastiCache Redis OSS](#) com Multi-AZ.
2. Em Cluster settings (Configurações do cluster)
 - a. Em Engine version (Versão do mecanismo), escolha a versão do mecanismo.
 - b. Em Port (Porta), use a porta padrão, 6379. Se você tiver um motivo para usar uma porta diferente, insira o número da porta.
 - c. Em Grupo de parâmetro (Parameter group), escolha um grupo de parâmetro ou crie um novo. Os grupo de parâmetros controlam os parâmetros de tempo de execução do seu cluster. Para obter mais informações sobre grupo de parâmetros, consulte [Parâmetros do Valkey e do Redis OSS](#) e [Criação de um grupo de parâmetros do ElastiCache](#).

 Note

Quando você seleciona um grupo de parâmetros para definir os valores de configuração do mecanismo, esse grupo de parâmetros é aplicado a todos os clusters no datastore global. Na página Parameter Groups (Grupos de parâmetros), o atributo Global indica se um grupo de parâmetros faz parte de um datastore global.

- d. Para Node type (Tipo de nó), escolha a seta para baixo (▼).

Na caixa de diálogo Change node type (Alterar tipo de nó), escolha um valor para a Instance family (Família de instâncias) para o tipo de nó desejado. Depois disso, escolha o tipo de nó que deseja usar para este cluster e escolha Save (Salvar).

Para obter mais informações, consulte [Escolha do tamanho do nó](#).

Se você escolher um tipo de nó r6gd, a classificação de dados em níveis será ativada automaticamente. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

- e. Em Número de fragmentos, escolha o número de fragmentos (partições/grupos de nós) desejado para este cluster do Valkey ou Redis OSS (modo cluster habilitado).

Para algumas versões do Valkey ou Redis OSS (modo cluster habilitado), é possível alterar dinamicamente o número de fragmentos no cluster:

- Redis OSS 3.2.10 e posteriores: se o cluster estiver executando o Redis OSS 3.2.10 ou versões posteriores, será possível alterar dinamicamente o número de fragmentos no cluster. Para obter mais informações, consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).
- Outras versões do Redis OSS: se o cluster estiver executando uma versão do Redis OSS anterior à versão 3.2.10, há outra abordagem. Para alterar o número de fragmentos no cluster nesse caso, crie um novo cluster com o novo número de fragmentos. Para obter mais informações, consulte [Restauração de um backup para um novo cache](#).

- f. Em Réplicas por fragmento, escolha o número de nós de réplica de leitura desejados em cada fragmento.

As restrições a seguir existem para o Valkey ou Redis OSS (modo cluster habilitado).

- Se você tiver o Multi-AZ habilitado, verifique se tem pelo menos uma réplica por fragmento.
- O número de réplicas é o mesmo para cada fragmento ao criar o cluster usando o console.
- O número de réplicas de leitura por fragmento é fixo e não pode ser alterado. Se você achar que precisa de mais ou menos réplicas por fragmento (API/CLI: grupo de nós), deverá criar um novo cluster com o novo número de réplicas. Para obter mais informações, consulte [Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente](#).

3. Em Connectivity (Conectividade)

- a. Em Network type (Tipo de rede), escolha as versões de IP compatíveis com esse cluster.
- b. Em Grupos de sub-redes, escolha a sub-rede que você deseja aplicar a esse cluster. ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. ElastiCache os clusters exigem uma sub-rede de pilha dupla com ambos IPv4 e IPv6 endereços atribuídos a eles para operar no modo de pilha dupla e uma IPv6 sub-rede somente para operar somente como -only. IPv6

Ao criar um novo grupo de sub-rede, insira o VPC ID (ID da VPC) ao qual ele pertence.

Selecione um Discovery IP type (Tipo de IP de descoberta). Somente os endereços IP do protocolo escolhido são retornados.

Para obter mais informações, consulte:

- [Escolhendo um tipo de rede em ElastiCache](#).
- [Criar uma sub-rede na VPC](#).

Se você for [Uso de zonas locais com o ElastiCache](#), deverá criar ou escolher uma sub-rede que esteja na região local.

Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).

4. Em Availability zone placements (Posicionamentos de zona de disponibilidade), você tem duas opções:

- Sem preferência — ElastiCache escolhe a Zona de Disponibilidade.
- Specify availability zones (Especificar zonas de disponibilidade): você especifica a zona de disponibilidade para cada cluster.

Se optar por especificar as zonas de disponibilidade, para cada cluster em cada fragmento, escolha a zona de disponibilidade na lista.

Para obter mais informações, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#).

5. Escolha Next (Próximo)
6. Em Configurações avançadas do Valkey ou Configurações avançadas do Redis OSS ou
 - Para Security (Segurança):
 - i. Para criptografar seus dados, você tem as seguintes opções:
 - Criptografia em repouso: permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

 Note

Você tem a opção de fornecer uma chave de criptografia diferente escolhendo a chave AWS KMS gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usar chaves gerenciadas pelo cliente do AWS KMS](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [criptografia em trânsito](#). Para o Valkey 7.2 e posteriores ou Redis OSS 6.0 e posteriores, se você habilitar a Criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Controle de acesso:
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.

- User Group Access Control List (Lista de controle de acesso de grupo de usuários): selecione um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para obter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
- Usuário padrão AUTH: um mecanismo de autenticação para um servidor Valkey ou Redis OSS. Para obter mais informações, consulte [AUTH](#).
- AUTH: um mecanismo de autenticação para servidores Valkey ou Redis OSS. Para obter mais informações, consulte [AUTH](#).

 Note

Para versões do Redis OSS 3.2.6 e posteriores, excluindo a versão 3.2.10, o AUTH é a única opção.

- ii. Em Grupos de segurança, escolha os grupos de segurança desejados para esse cluster. Um grupo de segurança atua como um firewall para controlar o acesso à rede ao cluster. É possível usar o grupo de segurança padrão para sua VPC ou criar um novo.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança para sua VPC](#) no Guia do usuário da Amazon VPC.

7. Para backups automáticos agendados regularmente, selecione Enable automatic backups (Habilitar backups automáticos) e insira o número de dias que deseja que cada backup automático seja mantido antes de ser excluído automaticamente. Se não quiser backups automáticos agendados regularmente, desmarque a caixa de seleção Enable automatic backups (Habilitar backups automáticos). Em ambos os casos, você sempre tem a opção de criar backups manuais.

Para obter mais informações sobre backup e restauração, consulte [Snapshots e restauração](#).

8. (Opcional) Especifique uma janela de manutenção. A opção janela de manutenção é o tempo, geralmente uma hora de duração, a cada semana quando o ElastiCache agenda a manutenção do sistema para seu cluster. Você pode permitir que o ElastiCache escolha o dia e a hora da sua janela de manutenção (No preference) ou pode escolher o dia, a hora e a duração por conta própria (Specify maintenance window). Se você escolher Especificar janela de manutenção, nas listas, escolha Dia de início, Hora de

início e Duração (em horas) para sua janela de manutenção. Todos os horários são em UCT.

Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

9. (Opcional) Para Logs:

- Em Log format (Formato do log), escolha Text (Texto) ou JSON.
- Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
- Em Destino do registro, escolha Criar novo e insira o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose, ou escolha Seleccionar existente e escolha o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose,

10. Para Tags, para ajudá-lo a gerenciar seus clusters e outros ElastiCache recursos, você pode atribuir seus próprios metadados a cada recurso na forma de tags. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).

11. Escolha Next (Próximo).

12. Revise todas as suas entradas e opções e faça as correções necessárias. Quando estiver pronto, escolha Create (Criar).

On premises

1. Para On-premises, recomendamos que você deixe Auto-failover (Failover automático) habilitado. Para obter mais informações, consulte [Minimizando o tempo de inatividade no ElastiCache Redis](#) OSS com Multi-AZ
2. Siga as etapas em [Uso do Outposts](#).

Para criar o equivalente usando a ElastiCache API ou AWS CLI em vez do ElastiCache console, veja o seguinte:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Assim que o status do seu cluster estiver disponível, você poderá conceder EC2 acesso a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4: Conectar-se ao nó de um cluster](#).

 Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Criar um cluster (AWS CLI)

Para criar um cluster usando o AWS CLI, use o `create-cache-cluster` comando.

Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Criação de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) (CLI)

Example– Um cluster do Valkey ou Redis OSS (modo cluster desabilitado) sem réplicas de leitura

O código da CLI a seguir cria um cluster do Valkey ou Redis OSS (modo cluster desabilitado) sem réplicas.

Note

Ao criar cluster usando um tipo de nó da família `r6gd`, é necessário repassar o parâmetro `data-tiering-enabled`.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--engine-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--cache-parameter-group default.redis6.x \  
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Para Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^
```

```
--engine redis ^  
--num-cache-nodes 1 ^  
--cache-parameter-group default.redis6.x ^  
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Criação de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (AWS CLI)

Clusters do Valkey ou Redis OSS (modo cluster habilitado) (API/CLI: grupos de replicação) não podem ser criados usando a operação `create-cache-cluster`. Para criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) (API/CLI: grupo de replicação), consulte [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster habilitado\) do início \(AWS CLI\)](#).

Para obter mais informações, consulte o tópico AWS CLI de ElastiCache referência [create-replication-group](#).

Criação de um cluster para Valkey ou Redis OSS (API) ElastiCache

Para criar um cluster usando a ElastiCache API, use a `CreateCacheCluster` ação.

Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver, mesmo que você não o esteja usando. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Tópicos

- [Criação de um cluster \(API\) Valkey ou Redis OSS \(modo de cluster desativado\) ElastiCache](#)
- [Criação de um cluster no Valkey ou no Redis OSS \(modo de cluster ativado\) \(API\) ElastiCache](#)

Criação de um cluster (API) Valkey ou Redis OSS (modo de cluster desativado) ElastiCache

O código a seguir cria um cluster (API) Valkey ou Redis OSS (modo de cluster desativado). ElastiCache

As quebras de linha foram adicionadas para legibilidade.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=CreateCacheCluster
&CacheClusterId=my-cluster
&CacheNodeType=cache.r4.large
&CacheParameterGroup=default.redis3.2
&Engine=redis
&EngineVersion=3.2.4
&NumCacheNodes=1
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&SnapshotArns.member.1=arn%3Aaws%3As3%3A%3A%3AmyS3Bucket%2Fdump.rdb
&Timestamp=20150508T220302Z
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20150508T220302Z
&X-Amz-Expires=20150508T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Signature=<signature>
```

Criação de um cluster no Valkey ou no Redis OSS (modo de cluster ativado) (API) ElastiCache

Clusters do Valkey ou Redis OSS (modo cluster habilitado) (API/CLI: grupos de replicação) não podem ser criados usando a operação `CreateCacheCluster`. Para criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) (API/CLI: grupo de replicação), consulte [Criação de um grupo de replicação no Valkey ou Redis OSS \(modo cluster habilitado\) do início \(API do ElastiCache\)](#).

Para obter mais informações, consulte o tópico de referência ElastiCache da API [CreateReplicationGroup](#).

Criação de um cluster do Memcached

Os exemplos a seguir mostram como criar um cluster usando Console de gerenciamento da AWS a ElastiCache API AWS CLI e.

Criação de um cluster do Memcached (console)

Quando você usa o mecanismo Memcached, a Amazon ElastiCache suporta o particionamento horizontal de seus dados em vários nós. O Memcached permite a descoberta automática e, portanto, você não precisa rastrear os endpoints de cada nó. O Memcached rastreia o endpoint de cada nó, atualizando a lista de endpoints à medida que os nós são adicionados e removidos. Seu aplicativo precisa apenas do endpoint de configuração para interagir com o cluster.

Para criar um cluster do Memcached por meio do console, siga as etapas em [Criação de um cluster do Valkey \(modo de cluster desabilitado\) \(console\)](#). Ao chegar à etapa cinco, selecione Criar cache do Memcached.

Assim que o status do seu cluster estiver disponível, você poderá conceder EC2 à Amazon acesso a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte as etapas semelhantes [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4: Conectar-se ao nó de um cluster](#).

 Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Criar um cluster (AWS CLI)

Para criar um cluster usando o AWS CLI, use o `create-cache-cluster` comando.

 Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver ativo, mesmo que você não o esteja usando ativamente. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Criar um cluster de cache Memcached (AWS CLI)

O código da CLI a seguir cria um cluster de Memcached com 3 nós.

Para 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
```

Para 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
```

Criação de um cluster para Memcached (API) ElastiCache

Para criar um cluster usando a ElastiCache API, use a `CreateCacheCluster` ação.

Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver, mesmo que você não o esteja usando. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluindo um cluster no ElastiCache](#).

Tópicos

- [Criação de um cluster Memcached \(API\) ElastiCache](#)

Criação de um cluster Memcached (API) ElastiCache

O código a seguir cria um cluster Memcached com 3 nós (ElastiCache API).

As quebras de linha foram adicionadas para legibilidade.

```
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>
```


Visualização dos detalhes de um cluster do ElastiCache

Você pode visualizar informações detalhadas sobre um ou mais clusters usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Visualização dos detalhes de um cluster do Memcached (console)

Você pode visualizar os detalhes de um cluster do Memcached usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

O procedimento a seguir detalha como visualizar os detalhes de um cluster do Memcached usando o console do ElastiCache.

Para visualizar os detalhes de um cluster Memcached

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS de interesse.
3. No painel do mecanismo ElastiCache, escolha Memcached. Isso exibe uma lista de todos os seus clusters que estão em execução no mecanismo Memcached.
4. Para ver os detalhes de um cluster, escolha a caixa de seleção à esquerda do nome do cluster.
5. Para visualizar as informações do nó, escolha a guia Nodes (Nós), que exibe informações sobre o status e o endpoint dos nós.
6. Para visualizar as métricas, escolha a guia Metrics (Métricas), que exibe as métricas relevantes para todos os nós do cluster. Para obter mais informações, consulte [Monitorando o uso com CloudWatch métricas](#).
7. Escolha a guia Network and security (Rede e segurança) para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-rede do cluster e o grupo de segurança da VPC. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).
8. Escolha a guia Maintenance (Manutenção) para ver detalhes sobre as configurações de manutenção do cluster. Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).
9. Escolha a guia Tags para ver detalhes sobre as tags aplicadas aos recursos do cluster. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).

Visualização dos detalhes do Valkey ou Redis OSS (modo cluster desabilitado) (console)

Você pode visualizar os detalhes de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

O procedimento a seguir detalha como visualizar os detalhes de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) usando o console do ElastiCache.

Para exibir os detalhes de um cluster do Valkey ou Redis OSS (modo cluster desabilitado)

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel do mecanismo ElastiCache, escolha Valkey ou Redis OSS para exibir uma lista de todos os seus clusters que estão sendo executados no mecanismo.
3. Para ver os detalhes de um cluster, marque a caixa de seleção à esquerda do nome do cluster. Certifique-se de selecionar um cluster que execute o mecanismo Valkey ou Redis OSS, não o Clustered Valkey ou o Clustered Redis OSS. Ao fazer isso, os detalhes sobre o cluster serão exibidos, incluindo o endpoint principal do cluster.
4. Como visualizar informações do nó:
 - a. Escolha o nome do cluster.
 - b. Selecione a guia Shards and nodes (Fragmentos e nós) Ao fazer isso, os detalhes sobre cada nó serão exibidos, incluindo o endpoint do nó, que você precisa usar para ler do cluster.
5. Para visualizar as métricas, escolha a guia Metrics (Métricas), que exibe as métricas relevantes para todos os nós do cluster. Para obter mais informações, consulte [Monitorando o uso com CloudWatch métricas](#)
6. Para visualizar os logs, escolha a guia Logs, que indica se o cluster está usando logs lentos ou logs do mecanismo e fornece detalhes relevantes. Para obter mais informações, consulte [Entrega de logs](#).
7. Escolha a guia Network and security (Rede e segurança) para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-redes do cluster. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).
8. Escolha a guia Maintenance (Manutenção) para ver detalhes sobre as configurações de manutenção do cluster. Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

9. Escolha a guia Service updates (Atualizações do serviço) para ver detalhes sobre todas as atualizações de serviço disponíveis, juntamente com a data de validade recomendada. Para obter mais informações, consulte [Atualizações de serviço em ElastiCache](#).
10. Escolha a guia Tags para ver detalhes sobre as tags aplicadas aos recursos do cluster. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).

Visualização de detalhes de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (console)

Você pode visualizar os detalhes de um cluster do Valkey ou Redis OSS (modo cluster habilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

O procedimento a seguir detalha como visualizar os detalhes de um cluster do Valkey ou Redis OSS (modo cluster habilitado) usando o console do ElastiCache.

Para exibir os detalhes de um cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS de interesse.
3. No painel do mecanismo ElastiCache, escolha Valkey ou Redis OSS para exibir uma lista de todos os seus clusters que estão sendo executados no mecanismo.
4. Para ver os detalhes de um cluster do Valkey ou Redis OSS (modo cluster habilitado), selecione a caixa à esquerda do nome do cluster. Certifique-se de escolher um cluster que esteja executando o mecanismo Valkey ou Clustered Redis OSS.

A tela se expande abaixo do cluster e exibe detalhes sobre o cluster, incluindo o endpoint de configuração de cluster.

5. Para ver uma lista dos fragmentos do cluster e o número de nós em cada fragmento, escolha a guia Shards and nodes (Fragmentos e nós).
6. Como visualizar informações específicas sobre um nó:
 - Escolha o ID do fragmento.

Isso exibirá informações sobre cada nó, incluindo o endpoint de cada nó que é necessário usar para ler dados do cluster.

7. Para visualizar as métricas, escolha a guia Metrics (Métricas), que exibe as métricas relevantes para todos os nós do cluster. Para obter mais informações, consulte [Monitorando o uso com CloudWatch métricas](#)
8. Para visualizar os logs, escolha a guia Logs, que indica se o cluster está usando logs lentos ou logs do mecanismo e fornece detalhes relevantes. Para obter mais informações, consulte [Entrega de logs](#).
9. Escolha a guia Network and security (Rede e segurança) para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-rede do cluster, o grupo de segurança da VPC e qual método de criptografia, se houver, está habilitado no cluster. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#) e [Segurança de dados no Amazon ElastiCache](#).
10. Escolha a guia Maintenance (Manutenção) para ver detalhes sobre as configurações de manutenção do cluster. Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).
11. Escolha a guia Service updates (Atualizações do serviço) para ver detalhes sobre todas as atualizações de serviço disponíveis, juntamente com a data de validade recomendada. Para obter mais informações, consulte [Atualizações de serviço em ElastiCache](#).
12. Escolha a guia Tags para ver detalhes sobre as tags aplicadas aos recursos do cluster. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).

Visualização dos detalhes de um cluster do ElastiCache (AWS CLI)

O seguinte código lista os detalhes de *my-cluster*:

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

Substitua *my-cluster* pelo nome de seu cluster em um caso em que o cluster seja criado com um nó de cache e nenhum fragmento usando o comando `create-cache-cluster`.

```
{
  "CacheClusters": [
    {
      "CacheClusterStatus": "available",
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
  "Engine": "redis",
  "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
  "CacheSubnetGroupName": "default",
  "SnapshotWindow": "08:30-09:30",
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false,
  "CacheClusterId": "my-cluster1",
  "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
  "PreferredAvailabilityZone": "us-west-2c",
  "AuthTokenEnabled": false,
  "PendingModifiedValues": {},
  "CacheNodeType": "cache.r4.large",
  "DataTiering": "disabled",
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "ParameterApplyStatus": "in-sync",
    "CacheParameterGroupName": "default.redis3.2"
  },
  "SnapshotRetentionLimit": 0,
  "AutoMinorVersionUpgrade": true,
  "EngineVersion": "3.2.10",
  "CacheSecurityGroups": [],
  "NumCacheNodes": 1
}

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
      "AuthTokenEnabled": false,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",

```

```

    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",

```

```

    "CacheClusterId": "my-cluster2-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2"
    },

```

```

    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.10",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
}

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "AuthTokenEnabled": true,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
      "CacheClusterStatus": "available",
      "AtRestEncryptionEnabled": true,
      "PreferredAvailabilityZone": "us-west-2a",
      "TransitEncryptionEnabled": true,
      "ReplicationGroupId": "my-cluster3",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
      "CacheClusterId": "my-cluster3-0001-001",
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",
      "DataTiering": "disabled",
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
      },
      "SnapshotRetentionLimit": 0,
      "EngineVersion": "6.0",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1
    },
  ],
}

```



```

{
  "SecurityGroups": [
    {
      "Status": "active",
      "SecurityGroupId": "sg-dbe93fa2"
    }
  ],
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
  "AuthTokenEnabled": true,
  "CacheSubnetGroupName": "default",
  "SnapshotWindow": "12:30-13:30",
  "AutoMinorVersionUpgrade": true,
  "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
  "CacheClusterStatus": "available",
  "AtRestEncryptionEnabled": true,
  "PreferredAvailabilityZone": "us-west-2b",
  "TransitEncryptionEnabled": true,
  "ReplicationGroupId": "my-cluster3",
  "Engine": "redis",
  "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
  "CacheClusterId": "my-cluster3-0001-002",
  "PendingModifiedValues": {},
  "CacheNodeType": "cache.r4.large",
  "DataTiering": "disabled",
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "ParameterApplyStatus": "in-sync",
    "CacheParameterGroupName": "default.redis3.2.cluster.on"
  },
  "SnapshotRetentionLimit": 0,
  "EngineVersion": "3.2.6",
  "CacheSecurityGroups": [],
  "NumCacheNodes": 1
},
{
  "SecurityGroups": [
    {
      "Status": "active",
      "SecurityGroupId": "sg-dbe93fa2"
    }
  ],
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",

```

```

    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,

```

```

    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],

```

```

        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
},
{
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
}
]

```

```
}
```

Em um caso em que o cluster seja criado com o Console de gerenciamento da AWS (nó de cluster habilitado ou desabilitado com um ou mais fragmentos), use o comando a seguir para descrever os detalhes do cluster [substitua *my-cluster* pelo nome do grupo de replicação (nome de seu cluster)]:

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Para obter mais informações, consulte a AWS CLI para o tópico do ElastiCache [describe-cache-clusters](#).

Visualização dos detalhes de um cluster do ElastiCache (API do ElastiCache)

Você pode visualizar os detalhes de um cluster usando a ação `DescribeCacheClusters` da API do ElastiCache. Se o parâmetro `CacheClusterId` estiver incluído, os detalhes do cluster especificado serão retornados. Se o parâmetro `CacheClusterId` for omitido, os detalhes para até `MaxRecords` (padrão 100) clusters serão retornados. O valor para `MaxRecords` não pode ser inferior a 20 ou superior a 100.

O código a seguir lista os detalhes para `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>
```

O código a seguir lista os detalhes para até 25 clusters.

```
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>
```

Para obter mais informações, consulte o tópico de referência da API do ElastiCache [DescribeCacheClusters](#).

Modificação de um cluster do ElastiCache

Além de adicionar ou remover nós de um cluster do ElastiCache, pode haver momentos em que você precisará fazer outras alterações, como adicionar um grupo de segurança, alterar a janela de manutenção ou um grupo de parâmetros.

Recomendamos que você tenha sua janela de manutenção cair no momento da menor utilização. Assim, talvez seja necessário modificá-la de tempos em tempos.

Quando você altera parâmetros de um cluster, a alteração é aplicada no cluster imediatamente ou após ele ser reiniciado. Isso é verdadeiro se você alterar o próprio grupo de parâmetro do cluster ou um valor do parâmetro dentro do grupo do parâmetro do cluster. Para determinar quando uma mudança de parâmetro particular é aplicada, consulte a seção *As alterações terão efeito da coluna Detalhes nas tabelas de* [Parâmetros específicos do Memcached](#) e [Parâmetros do Valkey e do Redis OSS](#). Para obter informações sobre como reinicializar um cluster, consulte [Reinicialização de nós](#).

Uso do Console de gerenciamento da AWS do ElastiCache

Como modificar um cluster

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS em que o cluster que você deseja modificar está localizado.
3. No painel de navegação, escolha o mecanismo em execução no cluster que deseja modificar.

É exibida uma lista dos clusters do mecanismo escolhido.

4. Na lista de clusters, no cluster que você deseja modificar, escolha seu nome.
5. Escolha Actions (Ações) e Modify (Modificar).

A janela Modify Cluster (Modificar cluster) é exibida.

6. Na janela Modify Cluster (Modificar cluster), faça as modificações desejadas. Entre as opções estão:
 - Descrição
 - Modo de cluster: para modificar o modo de cluster de Desativado para Ativado, você deve primeiro definir o modo de cluster como Compatível.

O modo compatível permite que seus clientes Valkey ou Redis OSS se conectem usando o modo de cluster habilitado e o modo de cluster desabilitado. Depois de migrar todos os clientes Valkey ou Redis OSS para usar o modo cluster habilitado, você poderá concluir a configuração do modo cluster e definir o modo cluster como Habilitado.

- Compatibilidade de versão de mecanismo

 Important

Você pode atualizar para versões mais recentes do mecanismo. Se você atualizar as principais versões do mecanismo, por exemplo, de 5.0.6 para 6.0, será necessário selecionar uma família de grupo de parâmetros que seja compatível com a nova versão do mecanismo. Para obter mais informações para fazer isso, consulte [Gerenciamento de versões para ElastiCache](#). No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

- Grupos de segurança da VPC
- Grupo do parâmetro
- Tipo de nó

 Note

Se o cluster estiver usando um tipo de nó da família r6gd, você só poderá escolher um tamanho de nó diferente nessa família. Se você escolher um tipo de nó da família r6gd, a classificação de dados em níveis será ativada automaticamente. Para mais informações, consulte [Classificação de dados em níveis](#).

- Multi-AZ
- Failover automático (somente no modo cluster desabilitado)
- Habilitar backups automáticos
- ID do nó de backup
- Backup Retention Period
- Janela de backup
- Tópico para notificação do SNS

- Compatibilidade de versão do mecanismo Memcached
- Tipo de rede

 Note

Se você estiver mudando de IPv4 para IPv6, deverá selecionar ou criar grupos de sub-rede compatíveis com IPv6. Para obter mais informações, consulte [Escolhendo um tipo de rede em ElastiCache](#).

- Grupos de segurança da VPC
- Grupo do parâmetro
- Janela de manutenção
- Tópico para notificação do SNS

A caixa Apply Immediately (Aplicar imediatamente) aplica-se apenas a modificações feitas no tipo de nó e na versão do mecanismo. Para aplicar alterações imediatamente, marque a caixa de seleção Apply Immediately (Aplicar imediatamente). Se essa caixa não for escolhida, as modificações feitas na versão do mecanismo serão aplicadas durante a próxima janela de manutenção. Outras modificações, como a alteração da janela de manutenção, são aplicadas imediatamente.

Para habilitar/desabilitar a entrega de logs para o Redis

1. Na lista de clusters, escolha o cluster que você deseja modificar. Selecione o Cluster name (Nome do cluster), e não a caixa de seleção ao lado dele.
2. Na página Detalhes do cluster, escolha a guia Logs.
3. Para habilitar/desabilitar logs lentos, escolha Habilitar ou Desabilitar.

Se você escolher habilitar:

- a. Em Log format (Formato do log), escolha JSON ou Text (Texto).
- b. Em Log destination type (Tipo de destino de log), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.

- c. Em Destino do log, escolha Criar novo e insira o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose. Você também pode escolher Selecionar existente e escolher o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose.
- d. Escolha Habilitar.

Para alterar sua configuração para o Redis:

1. Escolha Modificar.
2. Em Log format (Formato do log), escolha JSON ou Text (Texto).
3. Em Destination Type (Tipo de destino), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.
4. Em Log destination (Destino do log), escolha Create new (Criar) e insira o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose. Ou escolha Select existing (Selecionar existente) e escolha o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose.

Uso da AWS CLI com o ElastiCache

Você pode modificar um cluster existente usando a operação AWS CLI `modify-cache-cluster`. Para modificar o valor de configuração de um cluster, especifique o ID do cluster, o parâmetro a ser alterado e o novo valor do parâmetro. O exemplo a seguir altera a janela de manutenção para um cluster chamado `my-cluster` e aplica a alteração imediatamente.

Important

Você pode atualizar para versões mais recentes do mecanismo Memcached. Para obter mais informações para fazer isso, consulte [Gerenciamento de versões para ElastiCache](#). No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

Important

Você pode atualizar para versões mais recentes do mecanismo Valkey ou Redis OSS. Se você atualizar as principais versões do mecanismo, por exemplo, do Redis OSS 5.0.6 para

o Redis OSS 6.0, será preciso selecionar uma família de grupos de parâmetros que seja compatível com a nova versão do mecanismo. Para obter mais informações para fazer isso, consulte [Gerenciamento de versões para ElastiCache](#). No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

O parâmetro `--apply-immediately` aplica-se apenas a modificações no tipo de nó, na versão do mecanismo e a alteração do número de nós em um cluster. Se quiser aplicar qualquer uma dessas alterações imediatamente, use o parâmetro `--apply-immediately`. Se preferir adiar essas alterações para sua próxima janela de manutenção, use o parâmetro `--no-apply-immediately`. Outras modificações, como a alteração da janela de manutenção, são aplicadas imediatamente.

Para obter mais informações, consulte a AWS CLI para o tópico do ElastiCache [modify-cache-cluster](#).

Uso da API do ElastiCache

Você pode modificar um cluster existente usando a operação `ModifyCacheCluster` da API do ElastiCache. Para modificar o valor de configuração de um cluster, especifique o ID do cluster, o parâmetro a ser alterado e o novo valor do parâmetro. O exemplo a seguir altera a janela de manutenção para um cluster chamado `my-cluster` e aplica a alteração imediatamente.

Important

Você pode atualizar para versões mais recentes do mecanismo Memcached. Para obter mais informações para fazer isso, consulte [Gerenciamento de versões para ElastiCache](#).

No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

 Important

Você pode atualizar para versões mais recentes do mecanismo Valkey ou Redis OSS. Se você atualizar as principais versões do mecanismo, por exemplo, do Redis OSS 5.0.6 para o Redis OSS 6.0, será preciso selecionar uma família de grupos de parâmetros que seja compatível com a nova versão do mecanismo. Para obter mais informações para fazer isso, consulte [Gerenciamento de versões para ElastiCache](#). No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

As quebras de linha foram adicionadas para legibilidade.

```
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>
```

O parâmetro `ApplyImmediately` aplica-se apenas a modificações no tipo de nó, na versão do mecanismo e a alteração do número de nós em um cluster. Se quiser aplicar qualquer uma dessas alterações imediatamente, defina o parâmetro `ApplyImmediately` como `true`. Se preferir adiar essas alterações para sua próxima janela de manutenção, defina o parâmetro `ApplyImmediately` como `false`. Outras modificações, como a alteração da janela de manutenção, são aplicadas imediatamente.

Para obter mais informações, consulte o tópico de referência da API do ElastiCache [ModifyCacheCluster](#).

Adicionar nós a um ElastiCache cluster

Adicionar nós a um cluster Memcached aumenta o número de partições do seu cluster. Quando você altera o número de partições em um cluster, alguns dos principais espaços precisam ser remapeados para que sejam mapeados para o nó direito. Remapear espaços de chave aumenta temporariamente o número de falhas de cache no cluster. Para obter mais informações, consulte [Configurando seu ElastiCache cliente para balanceamento de carga eficiente \(Memcached\)](#).

Para reconfigurar o cluster do Valkey ou Redis OSS (modo cluster habilitado), consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#)

Você pode usar o ElastiCache Management Console AWS CLI ou a ElastiCache API para adicionar nós ao seu cluster.

Usando o ElastiCache Console de gerenciamento da AWS

Se você quiser adicionar um nó a um cluster do Valkey ou Redis OSS (modo cluster desabilitado) de nó único (sem replicação habilitada), o processo será de duas etapas: primeiro adicione a replicação e, depois, adicione um nó de réplica.

Tópicos

- [Para adicionar replicação a um cluster do Valkey ou Redis OSS sem fragmentos](#)
- [Para adicionar nós a um ElastiCache cluster \(console\)](#)

O procedimento a seguir adiciona replicação a um Valkey ou Redis OSS de nó único que não possui replicação habilitada. Quando você adiciona replicação, o nó existente torna-se o nó primário no cluster habilitado para replicação. Depois que a replicação é adicionada, você pode adicionar até 5 nós de réplica ao cluster.

Para adicionar replicação a um cluster do Valkey ou Redis OSS sem fragmentos

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.

Uma lista de clusters que executam o mecanismo Valkey ou Redis OSS é exibida.

3. Escolha o nome de um cluster (não a caixa à esquerda do nome do cluster) ao qual você quer adicionar nós.

O seguinte é verdade para um cluster do Redis OSS que não possui replicação habilitada:

- Ele está executando o Redis OSS, e não o Clustered Redis OSS.
- Tem zero estilhaços.

Se o cluster tiver estilhaços, a replicação já estará ativada e você poderá continuar em [Para adicionar nós a um ElastiCache cluster \(console\)](#).

4. Escolha Add replication.
5. Em Adicionar replicação, insira uma descrição para esse cluster habilitado para replicação.
6. Escolha Adicionar.

Assim que o status do cluster retornar available, você poderá continuar no próximo procedimento e adicionar réplicas ao cluster.

Para adicionar nós a um ElastiCache cluster (console)

O procedimento a seguir pode ser usado para adicionar nós a um cluster.

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha o mecanismo em execução no cluster ao qual você deseja adicionar nós.

É exibida uma lista de clusters que executam o mecanismo escolhido.

3. Na lista de clusters, no cluster ao qual você deseja adicionar um nó, escolha seu nome.

Se o cluster for um cluster do Valkey ou Redis OSS (modo cluster habilitado), consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).

Se o cluster é um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com zero fragmentos, primeiro conclua as etapas em [Para adicionar replicação a um cluster do Valkey ou Redis OSS sem fragmentos](#).

4. Escolha Adicionar nó.
5. Preencha as informações solicitadas na caixa de diálogo Add Node (Adicionar nó).

6. Escolha o botão Apply Immediately - Yes (Aplicar imediatamente - sim) para aplicar essa alteração imediatamente ou escolha No (Não) para adicionar este nó durante a próxima janela de manutenção do cluster.

Impacto de novas solicitações de adição e remoção em solicitações pendentes

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 1	Delete	Delete	A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de exclusão pendente. Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para excluir os nós 0002 e 0004 for emitida, somente os nós 0002 e 0004 serão excluídos. Os nós 0001, 0003 e 0007 não serão excluídos.
Cenário 2	Delete	Criar	A nova solicitação de criação, pendente ou imediata, substitui a solicitação de exclusão pendente. Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para criar um nó for emitida, um novo nó será criado e os nós 0001, 0003 e 0007 não serão excluídos.
Cenário 3	Criar	Delete	A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de criação pendente. Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para excluir o nó 0003, nenhum novo nó será criado e o nó 0003 será excluído.

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 4	Criar	Criar	A nova solicitação de criação é adicionada à solicitação de criação pendente. Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para criar três nós, as novas solicitações serão adicionadas à solicitação pendente e cinco nós serão criados.  Important Se a nova solicitação de criação estiver definida como Apply immediately - Yes (Aplicar imediatamente - Sim), todas as solicitações de criação são executadas imediatamente. Se a nova solicitação de criação estiver definida como Apply immediately - No (Aplicar imediatamente - No), todas as solicitações de criação ficam pendentes.

Para determinar quais operações estão pendentes, escolha a opção Description (Descrição) e verifique para ver quantas criações ou exclusões pendentes são mostradas. Você não pode ter criações pendentes e exclusões pendentes.

7. Escolha o botão Add.

Depois de alguns momentos, os novos nós deverão aparecer na lista de nós com um status de creating. Se não aparecem, atualize a página do navegador. Quando o status mudar para available do nó do novo nó pode ser usado.

Usando o AWS CLI com ElastiCache

Para adicionar nós a um cluster usando o AWS CLI, use a AWS CLI operação `modify-cache-cluster` com os seguintes parâmetros:

- `--cache-cluster-id`: o ID do cluster ao qual você deseja adicionar os nós.
- `--num-cache-nodes`: o parâmetro `--num-cache-nodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada. Para adicionar nós a esse cluster, `--num-cache-nodes` deve ser maior que o número atual de nós neste cluster. Se esse valor for menor que o número atual de nós, ElastiCache espera que o parâmetro `cache-node-ids-to-remove` e uma lista de nós sejam removidos do cluster. Para obter mais informações, consulte [Usando o AWS CLI com ElastiCache](#).
- `--apply-immediately` ou `--no-apply-immediately`, que especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",
```

```

    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
        "Port": 11211,
        "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "NumCacheNodes": 2,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
        "NumCacheNodes": 5
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
}
}

```

Para obter mais informações, consulte o AWS CLI tópico [modify-cache-cluster](#).

Usando o AWS CLI com ElastiCache

Se quiser adicionar nós a um cluster existente do Valkey ou Redis OSS (modo cluster desabilitado) que não tenha a replicação habilitada, você deverá primeiro criar o grupo de replicação especificando o cluster existente como primário. Para obter mais informações, consulte [Criação de um grupo de replicação usando um cluster do Valkey ou Redis OSS disponível \(AWS CLI\)](#). Depois que o grupo de replicação estiver disponível, você poderá continuar com o processo a seguir.

Para adicionar nós a um cluster usando o AWS CLI, use a AWS CLI operação `increase-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: o ID do grupo de replicação ao qual você deseja adicionar nós.

- `--new-replica-count`: especifica o número de nós que você deseja neste grupo de replicação após a modificação ser aplicada. Para adicionar nós a esse cluster, `--new-replica-count` deve ser maior que o número atual de nós neste cluster.
- `--apply-immediately` ou `--no-apply-immediately`, que especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.

Para Linux, macOS ou Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 4 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 4 ^  
  --apply-immediately
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "node-test-001",  
      "node-test-002",  
      "node-test-003",  
      "node-test-004",  
      "node-test-005"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
```

```

        "Port": 6379
    },
    "ReaderEndpoint": {
        "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "NodeGroupMembers": [
        {
            "CacheClusterId": "node-test-001",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "primary"
        },
        {
            "CacheClusterId": "node-test-002",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2c",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "node-test-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2b",
            "CurrentRole": "replica"
        }
    ]
},
"SnapshottingClusterId": "node-test-002",

```

```
    "AutomaticFailover": "enabled",
    "MultiAZ": "enabled",
    "SnapshotRetentionLimit": 1,
    "SnapshotWindow": "07:30-08:30",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r5.large",
    "DataTiering": "disabled",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false,
    "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
  }
}
```

Para obter mais informações, consulte o AWS CLI tópico [increase-replica-count](#).

Usando a ElastiCache API

Se quiser adicionar nós a um cluster existente do Valkey ou Redis OSS (modo cluster desabilitado) que não tenha a replicação habilitada, você deverá primeiro criar o grupo de replicação especificando o cluster existente como primário. Para obter mais informações, consulte [Adicionar réplicas a um cluster \(API\) autônomo Valkey ou Redis OSS \(modo de cluster desativado\) ElastiCache](#). Depois que o grupo de replicação estiver disponível, você poderá continuar com o processo a seguir.

Para adicionar nós a um cluster (ElastiCache API)

- Chame a operação de API `IncreaseReplicaCount` com os seguintes parâmetros:
 - `ReplicationGroupId`: o ID do cluster ao qual você deseja adicionar os nós.
 - `NewReplicaCount`: o parâmetro `NewReplicaCount` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada. Para adicionar nós a esse cluster, `NewReplicaCount` deve ser maior que o número atual de nós neste cluster. Se esse valor for inferior ao número atual de nós, use a API `DecreaseReplicaCount` com o número de nós a serem removidos do cluster.
 - `ApplyImmediately` Especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.
 - `Region` Especifica a AWS região do cluster à qual você deseja adicionar nós.

O exemplo a seguir mostra uma chamada para adicionar nós a um cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=true  
  &NumCacheNodes=4  
  &ReplicationGroupId=my-replication-group  
  &Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico ElastiCache da API [IncreaseReplicaCount](#).

Usando a ElastiCache API

Para adicionar nós a um cluster (ElastiCache API)

- Chame a operação de API `ModifyCacheCluster` com os seguintes parâmetros:
 - `CacheClusterId`: o ID do cluster ao qual você deseja adicionar os nós.
 - `NumCacheNodes`: o parâmetro `NumCacheNodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada. Para adicionar nós a esse cluster, `NumCacheNodes` deve ser maior que o número atual de nós neste cluster. Se esse valor for menor que o número atual de nós, ElastiCache espera que o parâmetro `CacheNodeIdsToRemove` com uma lista de nós seja removido do cluster (consulte [Usando a ElastiCache API com o Memcached](#)).
 - `ApplyImmediately` Especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.
 - `Region` Especifica a AWS região do cluster à qual você deseja adicionar nós.

O exemplo a seguir mostra uma chamada para adicionar nós a um cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &NumCacheNodes=5  
&CacheClusterId=my-cluster  
&Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico ElastiCache da API [ModifyCacheCluster](#).

Removendo nós de um ElastiCache cluster

Você pode excluir um nó de um cluster OSS Valkey, Memcached ou Redis usando a Console de gerenciamento da AWS, a ou a AWS CLI API. ElastiCache

Note

Cada vez que você altera o número de nós em um cluster Memcached, você deve remapear pelo menos alguns dos seus espaços de chaves para que ele seja mapeado para o nó correto. Para obter informações mais detalhadas sobre o balanceamento de carga de um cluster Memcached, consulte [Configurando seu ElastiCache cliente para balanceamento de carga eficiente \(Memcached\)](#).

Usando o ElastiCache Console de gerenciamento da AWS

Para remover nós de um cluster (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região do cluster da qual você deseja remover os nós.
3. No painel de navegação, escolha o mecanismo em execução no cluster do qual você deseja remover um nó.

É exibida uma lista de clusters que executam o mecanismo escolhido.

4. Na lista de clusters, escolha o nome do cluster do qual você deseja remover um nó.

É exibida uma lista dos nós do cluster.

5. Escolha a caixa à esquerda do ID do nó para o nó que você deseja remover. Usando o console do ElastiCache, é possível excluir somente um nó de cada vez. Portanto, escolher vários nós significa que não é possível usar o botão Excluir nó.

A página Excluir nó é exibida.

6. Para excluir o nó, preencha a página Excluir nó e escolha Excluir nó. Para manter o nó, escolha Cancelar.

⚠ Important

Com o Valkey ou Redis OSS, se você estiver excluindo os resultados do nó no cluster que não são mais compatíveis com Multi-AZ, certifique-se primeiro de desmarcar a caixa de seleção Multi-AZ e depois excluir o nó. Se você desmarcar a caixa de seleção Multi-AZ, será possível optar por habilitar o Failover automático.

Impacto de novas solicitações de adição e remoção em solicitações pendentes

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 1	Delete	Delete	A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de exclusão pendente. Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para excluir os nós 0002 e 0004 for emitida, somente os nós 0002 e 0004 serão excluídos. Os nós 0001, 0003 e 0007 não serão excluídos.
Cenário 2	Delete	Criar	A nova solicitação de criação, pendente ou imediata, substitui a solicitação de exclusão pendente. Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para criar um nó for emitida, um novo nó será criado e os nós 0001, 0003 e 0007 não serão excluídos.
Cenário 3	Criar	Delete	A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de criação pendente. Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para excluir o nó 0003, nenhum novo nó será criado e o nó 0003 será excluído.

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 4	Criar	Criar	A nova solicitação de criação é adicionada à solicitação de criação pendente. Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para criar três nós, as novas solicitações serão adicionadas à solicitação pendente e cinco nós serão criados.  Important Se a nova solicitação de criação estiver definida como Apply immediately - Yes (Aplicar imediatamente - Sim), todas as solicitações de criação são executadas imediatamente. Se a nova solicitação de criação estiver definida como Apply immediately - No (Aplicar imediatamente - No), todas as solicitações de criação ficam pendentes.

Para determinar quais operações estão pendentes, escolha a opção Description (Descrição) e verifique para ver quantas criações ou exclusões pendentes são mostradas. Você não pode ter criações pendentes e exclusões pendentes.

Usando o AWS CLI com ElastiCache

1. Identifique IDs os nós que você deseja remover. Para obter mais informações, consulte [Visualização dos detalhes de um cluster do ElastiCache](#).
2. Use a operação `decrease-replica-count` da CLI com uma lista dos nós a serem removidos, como no exemplo a seguir.

Para remover nós de um cluster usando a interface da linha de comando, use o comando `decrease-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: o ID do grupo de replicação do qual você deseja remover nós.
- `--new-replica-count`: o parâmetro `--new-replica-count` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada.
- `--replicas-to-remove` Uma lista de nós IDs que você deseja remover desse cluster.
- `--apply-immediately` ou `--no-apply-immediately` especifica se você deve remover esses nós imediatamente ou na próxima janela de manutenção.
- `--region` Especifica a AWS região do cluster da qual você deseja remover os nós.

Note

Você pode passar apenas um dos parâmetros `--replicas-to-remove` ou `--new-replica-count` ao chamar esta operação.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 2 \  
  --region us-east-2 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 3 ^  
  --region us-east-2 ^  
  --apply-immediately
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test"  
  },  
}
```

```
"Status": "modifying",
"PendingModifiedValues": {},
"MemberClusters": [
    "node-test-001",
    "node-test-002",
    "node-test-003",
    "node-test-004",
    "node-test-005",
    "node-test-006"
],
"NodeGroups": [
    {
        "NodeGroupId": "0001",
        "Status": "modifying",
        "PrimaryEndpoint": {
            "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "ReaderEndpoint": {
            "Address": "node-test-
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "NodeGroupMembers": [
            {
                "CacheClusterId": "node-test-001",
                "CacheNodeId": "0001",
                "ReadEndpoint": {
                    "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "PreferredAvailabilityZone": "us-west-2a",
                "CurrentRole": "primary"
            },
            {
                "CacheClusterId": "node-test-002",
                "CacheNodeId": "0001",
                "ReadEndpoint": {
                    "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "PreferredAvailabilityZone": "us-west-2c",
```

```
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-004",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-005",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-006",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
```

```

        "CurrentRole": "replica"
      }
    ]
  },
  "SnapshottingClusterId": "node-test-002",
  "AutomaticFailover": "enabled",
  "MultiAZ": "enabled",
  "SnapshotRetentionLimit": 1,
  "SnapshotWindow": "07:30-08:30",
  "ClusterEnabled": false,
  "CacheNodeType": "cache.r5.large",
  "DataTiering": "disabled",
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false,
  "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
}
}

```

Alternativamente, você também poderia chamar o `decrease-replica-count` e, em vez de passar o parâmetro `--new-replica-count`, você pode passar o parâmetro `--replicas-to-remove`, como mostrado a seguir:

Para Linux, macOS ou Unix:

```

aws elasticache decrease-replica-count \
  --replication-group-id my-replication-group \
  --replicas-to-remove node-test-003 \
  --region us-east-2 \
  --apply-immediately

```

Para Windows:

```

aws elasticache decrease-replica-count ^
  --replication-group-id my-replication-group ^
  --replicas-to-remove node-test-003 ^
  --region us-east-2 ^
  --apply-immediately

```

Para obter mais informações, consulte os AWS CLI tópicos [decrease-replica-count](#).

Usando a ElastiCache API com Valkey ou Redis OSS

Para remover nós usando a ElastiCache API, chame a operação da `DecreaseReplicaCount` API com o ID do grupo de replicação e uma lista de nós a serem removidos, conforme mostrado:

- `ReplicationGroupId`: o ID do grupo de replicação do qual você deseja remover nós.
- `ReplicasToRemove`: o parâmetro `ReplicasToRemove` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada.
- `ApplyImmediately` Especifica se você deve remover esses nós imediatamente ou na próxima janela de manutenção.
- `Region` Especifica a AWS região do cluster da qual você deseja remover um nó.

O exemplo a seguir remove imediatamente os nós 0004 e 0005 do cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DecreaseReplicaCount  
  &ReplicationGroupId=my-replication-group  
  &ApplyImmediately=true  
  &ReplicasToRemove=node-test-003  
  &Region us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico ElastiCache da API [DecreaseReplicaCount](#).

Usando a ElastiCache API com o Memcached

Para remover nós usando a ElastiCache API, chame a operação da `ModifyCacheCluster` API com o ID do cluster de cache e uma lista de nós a serem removidos, conforme mostrado:

- `CacheClusterId` o ID do cluster de cache do qual você deseja remover nós.

- `NumCacheNodes`: o parâmetro `NumCacheNodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada.
- `CacheNodeIdsToRemove.member.n` A lista de nós IDs a serem removidos do cluster.
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` Especifica se você deve remover esses nós imediatamente ou na próxima janela de manutenção.
- `Region` Especifica a AWS região do cluster da qual você deseja remover um nó.

O exemplo a seguir remove imediatamente os nós 0004 e 0005 do 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>
```

Para obter mais informações, consulte o tópico ElastiCache da API [ModifyCacheCluster](#).

Cancelamento de operações de adição ou exclusão de nós pendentes no ElastiCache

Se você tiver optado por não aplicar uma alteração de cluster do ElastiCache imediatamente, a operação terá o status pendente até que seja realizada na sua próxima janela de manutenção. Você pode cancelar qualquer operação pendente de adição ou exclusão.

Para cancelar uma operação pendente de adição ou exclusão de nó com o AWS CLI, use o comando `modify-cache-cluster`. Defina `num-cache-nodes` igual ao número atual de nós de cache no cluster e, em seguida, adicione o sinalizador `--apply-immediately`. Isso substituirá a alteração pendente.

Para cancelar a adição ou exclusão de um nó pendente:

```
aws elasticache modify-cache-cluster
--cache-cluster-id <your-cluster-id>
--num-cache-nodes <current-number-of-nodes>
--apply-immediately
```

Se não estiver claro se alguma adição ou exclusão de nó está pendente, você pode confirmar seu status com o comando `describe-cache-clusters`:

```
aws elasticache describe-cache-clusters
--cache-cluster-id <your-cluster-id>
```

Todos os nós pendentes devem aparecer na saída `PendingModifiedValues`. Por exemplo:

```
"PendingModifiedValues": {
  "NumCacheNodes": 3
},
```

Excluindo um cluster no ElastiCache

Enquanto um ElastiCache cluster estiver no estado disponível, você será cobrado por ele, esteja ou não o usando ativamente. Para interromper as cobranças, exclua o cluster.

Warning

Quando você exclui um ElastiCache cluster, seus instantâneos manuais são retidos. Também é possível criar um snapshot final antes que o cluster seja excluído. Os snapshots de cache automáticos não são retidos.

Usando o Console de gerenciamento da AWS

O procedimento a seguir exclui um único cluster da sua implantação. Para excluir vários clusters, repita o procedimento para cada cluster que deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster antes de iniciar o procedimento para excluir outro.

Para excluir um cluster

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache mecanismo, escolha o mecanismo que está sendo executado no cluster que você deseja excluir.

É exibida uma lista de todos os clusters que executam esse mecanismo.

3. Para escolher o cluster a ser excluído, escolha o nome na lista de clusters.

Important

Você só pode excluir um cluster por vez do ElastiCache console. Escolher vários clusters desabilita a operação de exclusão.

4. Em Ações, escolha Excluir.
5. Na tela de confirmação Excluir cluster, escolha Excluir para excluir o cluster ou Cancelar para mantê-lo.

Se você escolheu Excluir, o status do cluster muda para excluindo.

Assim que o cluster não estiver mais relacionado na lista de clusters, você para de ser cobrado por ele.

Usando o AWS CLI para excluir um ElastiCache cluster

O código a seguir exclui o ElastiCache cluster `my-cluster`.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

A ação `delete-cache-cluster` da CLI apenas exclui um cluster. Para excluir vários clusters, chame `delete-cache-cluster` para cada cluster de cache que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster antes de excluir outro.

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

Para Windows:

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

Para obter mais informações, consulte o ElastiCache tópico AWS CLI for [delete-cache-cluster](#).

Usando a ElastiCache API

O código a seguir exclui o 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>
```

A operação `DeleteCacheCluster` da API exclui apenas um cluster. Para excluir vários clusters, chame `DeleteCacheCluster` para cada cluster que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster antes de excluir outro.

Para obter mais informações, consulte o tópico de referência ElastiCache da API [DeleteCacheCluster](#).

Acessando seu ElastiCache cluster ou grupo de replicação

Suas ElastiCache instâncias da Amazon foram projetadas para serem acessadas por meio de uma EC2 instância da Amazon.

Se você lançou sua ElastiCache instância em uma Amazon Virtual Private Cloud (Amazon VPC), você pode acessar sua ElastiCache instância de uma instância da Amazon na EC2 mesma Amazon VPC. Ou, usando o emparelhamento de VPC, você pode acessar sua ElastiCache instância de uma Amazon EC2 em uma Amazon VPC diferente.

Se você lançou sua ElastiCache instância no EC2 Classic, você permite que a EC2 instância acesse seu cluster concedendo ao grupo de EC2 segurança da Amazon associado à instância acesso ao seu grupo de segurança de cache. Por padrão, o acesso a um cluster é restrito à conta que o executou.

Tópicos

- [Concessão de acesso ao cluster ou grupo de replicação](#)

Concessão de acesso ao cluster ou grupo de replicação

Você lançou seu cluster em EC2 -VPC

Se você lançou seu cluster em uma Amazon Virtual Private Cloud (Amazon VPC), você pode se conectar ao seu ElastiCache cluster somente a partir de uma EC2 instância da Amazon que esteja sendo executada na mesma Amazon VPC. Nesse caso, você precisará conceder entrada de rede ao cluster.

Note

Se você estiver usando Local Zones (Zonas locais), verifique se você as habilitou. Para obter mais informações, consulte [Habilitar zonas locais](#). Ao fazer isso, sua VPC é estendida para essa zona local e sua VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade, e gateways relevantes, tabelas de rota e outras considerações de grupos de segurança serão ajustados automaticamente.

Para conceder entrada na rede de um grupo de segurança da Amazon VPC para um cluster

1. Faça login no Console de gerenciamento da AWS e abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação, em Network & Security, escolha Security Groups.
3. Na lista de grupos de segurança, escolha o de segurança para a sua Amazon VPC. A menos que você tenha criado um grupo de segurança para ElastiCache uso, esse grupo de segurança será denominado padrão.
4. Escolha a guia Inbound e faça o seguinte:
 - a. Escolha Editar.
 - b. Escolha Adicionar regra.
 - c. Na coluna Tipo, escolha Regra TCP personalizada.
 - d. Na caixa Port range, digite o número da porta para o nó do cluster. Esse número deve ser o mesmo que você especificou quando você executou o cluster. A porta padrão do Memcached é **11211** A porta padrão do Valkey e Redis OSS é **6379**.
 - e. Na caixa Fonte, escolha Qualquer lugar que tenha o intervalo de portas (0.0.0.0/0) para que qualquer EC2 instância da Amazon que você iniciar na sua Amazon VPC possa se conectar aos seus nós. ElastiCache

 Important

Abrir o ElastiCache cluster para 0.0.0.0/0 não o expõe à Internet porque ele não tem endereço IP público e, portanto, não pode ser acessado de fora da VPC. No entanto, o grupo de segurança padrão pode ser aplicado a outras EC2 instâncias da Amazon na conta do cliente, e essas instâncias podem ter um endereço IP público. Se eles estiverem executando algo na porta padrão, esse serviço poderá ser exposto involuntariamente. Portanto, recomendamos criar um grupo de segurança de VPC que será usado exclusivamente pelo ElastiCache. Para obter mais informações, consulte [Grupos de segurança personalizados](#).

- f. Escolha Salvar.

Quando você executa uma EC2 instância da Amazon em sua Amazon VPC, essa instância poderá se conectar ao seu ElastiCache cluster.

Acessando ElastiCache recursos de fora AWS

A Amazon ElastiCache é um AWS serviço que fornece armazenamento de valores-chave na memória baseado em nuvem. O serviço foi projetado para ser acessado exclusivamente de dentro AWS. No entanto, se o ElastiCache cluster estiver hospedado em uma VPC, você poderá usar uma instância de Network Address Translation (NAT) para fornecer acesso externo.

Requisitos

Os seguintes requisitos devem ser atendidos para que você possa acessar seus ElastiCache recursos de fora AWS:

- O cluster deve residir em uma VPC e ser acessado por meio de uma instância NAT (Conversão de endereços de rede). Não existem exceções para esse requerimento.
- A instância NAT deve ser executada na mesma VPC que o cluster.
- A instância NAT deve ser iniciada em uma sub-rede pública separada do cluster.
- Um endereço IP elástico (EIP) deve ser associado à instância NAT. O recurso de encaminhamento de porta de iptables é usado para encaminhar uma porta na instância NAT à porta do nó de cache na VPC.

Considerações

As seguintes considerações devem ser mantidas em mente ao acessar seus recursos do ElastiCache de fora do ElastiCache.

- Os clientes se conectam ao EIP e à porta de cache da instância NAT. O encaminhamento de porta na instância NAT direciona o tráfego ao nó do cluster apropriado.
- Se um nó de cluster for adicionado ou substituído, as regras de iptables precisarão ser atualizadas para refletir essa alteração.

Limitações

Essa abordagem deve ser usada somente para fins de teste e desenvolvimento. Ela não é recomendada para uso em produção devido às seguintes limitações:

- A instância NAT está atuando como um proxy entre clientes e vários clusters. A adição de um proxy afeta o desempenho do cluster. O impacto aumenta com o número de clusters que você está acessando por meio da instância NAT.

- O tráfego dos clientes para a instância NAT não é criptografado. Portanto, você deve evitar o envio de dados confidenciais através da instância NAT.
- A instância NAT acrescenta a sobrecarga de manter outra instância.
- A instância NAT serve como um único ponto de falha. Para obter informações sobre como configurar o NAT de alta disponibilidade na VPC, consulte [Alta disponibilidade para instâncias NAT da Amazon VPC: um exemplo](#).

Como acessar ElastiCache recursos de fora AWS

O procedimento a seguir demonstra como se conectar aos seus ElastiCache recursos usando uma instância NAT.

Estas etapas assumem o seguinte:

- `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`

Em seguida, você precisa da NAT na direção oposta:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

Você também precisa habilitar o encaminhamento de IP, que permanece desabilitado por padrão:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf  
sudo sysctl --system
```

- Você está acessando um cluster Memcached com:
 - Endereço IP: 10.0.1.230
 - Porta Memcached padrão: 11211
 - Grupo de segurança: *10\0\0\0\55*
- Você está acessando um cluster do Valkey ou Redis OSS com:
 - Endereço IP: 10.0.1.230
 - Porta padrão: 6379
 - Grupo de segurança: sg-bd56b7da

- AWS endereço IP da instância — 198.99.100.27
- Seu cliente confiável possui o endereço IP 198.51.100.27.
- Sua instância NAT possui o endereço IP elástico 203.0.113.73.
- Sua instância NAT tem o security group sg-ce56b7a9.

Para se conectar aos seus ElastiCache recursos usando uma instância NAT

1. Crie uma instância NAT na mesma VPC que o seu cluster, mas em uma sub-rede pública.

Por padrão, o assistente da VPC iniciará um tipo de nó cache.m1.small. Você deve selecionar um tamanho de nó conforme as suas necessidades. Você deve usar o EC2 NAT AMI para poder acessar ElastiCache de fora AWS.

Para obter informações sobre como criar uma instância NAT, consulte [Instâncias NAT no Guia do usuário da AWS VPC](#).

2. Crie regras de grupo de segurança para o cluster e a instância NAT.

O grupo de segurança da instância NAT e a instância do cluster devem ter as seguintes regras:

- Duas regras de entrada
 - Com o Memcached, a primeira regra é permitir conexões TCP de clientes confiáveis com cada porta de cache reencaminhada da instância NAT (11211 a 11213).
 - Com Valkey e Redis OSS, a primeira regra é permitir conexões TCP de clientes confiáveis com cada porta de cache reencaminhada da instância NAT (6379 a 6381).
- Uma segunda regra para permitir acesso SSH a clientes confiáveis.

Grupo de segurança da instância NAT: regras de entrada com o Memcached

Tipo	Protocolo	Intervalo de portas	Fonte
Regra personalizada de TCP	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

Grupo de segurança da instância NAT: regras de entrada com Valkey ou Redis OSS

Tipo	Protocolo	Intervalo de portas	Fonte
Regra personalizada de TCP	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Com o Memcached, uma regra de saída para permitir conexões TCP para a porta de cache (11211).

Grupo de segurança da instância NAT: regra de saída

Tipo	Protocolo	Intervalo de portas	Destino
Regra personalizada de TCP	TCP	11211	sg-ce56b7a9 (grupo de segurança de instância NAT)

- Com o Valkey ou Redis OSS, uma regra de saída para permitir conexões TCP para a porta de cache (6379).

Grupo de segurança da instância NAT: regra de saída

Tipo	Protocolo	Intervalo de portas	Destino
Regra personalizada de TCP	TCP	6379	sg-ce56b7a9 (grupo de segurança de instância NAT)

- Com o Memcached, uma regra de entrada para o grupo de segurança do cluster que permite conexões TCP da instância NAT com a porta de cache (11211).

Grupo de segurança da instância do cluster: regra de entrada

Tipo	Protocolo	Intervalo de portas	Fonte
Regra personalizada de TCP	TCP	11211	sg-ce56b7a9 (grupo de segurança de instância NAT)

- Com o Valkey ou Redis OSS, uma regra de entrada para o grupo de segurança do cluster que permite conexões TCP da instância NAT com a porta de cache (6379).

Grupo de segurança da instância do cluster: regra de entrada

Tipo	Protocolo	Intervalo de portas	Fonte
Regra personalizada de TCP	TCP	6379	sg-ce56b7a9 (grupo de segurança de instância NAT)

3. Valide as regras.

- Confirme se o cliente confiável é capaz de fazer SSH para a instância NAT.
- Confirme se o cliente confiável pode se conectar ao cluster a partir da instância NAT.

4. Memcached

Adicione uma regra iptables à instância NAT.

Uma regra iptables deve ser adicionada à tabela NAT para cada nó no cluster para encaminhar a porta de cache da instância NAT ao nó do cluster. Um exemplo pode ser o seguinte:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
```

O número da porta deve ser exclusivo para cada nó no cluster. Por exemplo, se estiver trabalhando com um cluster Memcached de três nós usando as portas 11211 - 11213, as regras seriam as seguintes:

```
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
```

Confirme se o cliente confiável pode se conectar ao cluster.

O cliente confiável deve se conectar ao EIP associado à instância NAT e à porta de cluster correspondente ao nó de cluster apropriado. Por exemplo, a string de conexão para o PHP pode ser o seguinte:

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

Um cliente telnet também pode ser usado para verificar a conexão. Por exemplo:

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

Valkey ou Redis OSS

Adicione uma regra iptables à instância NAT.

Uma regra iptables deve ser adicionada à tabela NAT para cada nó no cluster para encaminhar a porta de cache da instância NAT ao nó do cluster. Um exemplo pode ser o seguinte:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
```

O número da porta deve ser exclusivo para cada nó no cluster. Por exemplo, se estiver trabalhando com um cluster do Redis OSS de três nós usando as portas 6379 a 6381, as regras seriam as seguintes:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to
10.0.1.231:6379
```

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379
```

Confirme se o cliente confiável pode se conectar ao cluster.

O cliente confiável deve se conectar ao EIP associado à instância NAT e à porta de cluster correspondente ao nó de cluster apropriado. Por exemplo, a string de conexão para o PHP pode ser o seguinte:

```
redis->connect( '203.0.113.73', 6379 );  
redis->connect( '203.0.113.73', 6380 );  
redis->connect( '203.0.113.73', 6381 );
```

Um cliente telnet também pode ser usado para verificar a conexão. Por exemplo:

```
telnet 203.0.113.73 6379  
telnet 203.0.113.73 6380  
telnet 203.0.113.73 6381
```

5. Salve a configuração iptables.

Salve as regras depois de testá-las e verificá-las. Se você estiver usando uma distribuição Linux baseada no Redhat (como o Amazon Linux), execute o seguinte comando:

```
service iptables save
```

Tópicos relacionados

Os tópicos a seguir podem ser de seu interesse.

- [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#)
- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente](#)
- [Instâncias NAT](#)
- [Configurando clientes ElastiCache](#)
- [Alta disponibilidade para instâncias NAT da Amazon VPC: um exemplo](#)

Encontrar endpoints de conexão no ElastiCache

Sua aplicação conecta-se ao seu cluster do ElastiCache usando endpoints. Um endpoint é o endereço exclusivo de um nó ou cluster.

Também é possível estabelecer uma conexão privada entre a VPC e os endpoints da API do ElastiCache criando um endpoint da VPC da interface por meio do AWS PrivateLink. Para obter mais informações, consulte [ElastiCache API e interface VPC endpoints \(AWS PrivateLink\)](#).

Quais endpoints usar com o Valkey ou Redis OSS.

- Para o nó autônomo, use o endpoint do nó para operações de leitura e gravação.
- Para clusters do Valkey ou Redis OSS (modo cluster desabilitado), use o Endpoint primário para todas as operações de gravação. Use o Endpoint de leitor para dividir uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura. Use os Endpoints de nó individuais para operações de leitura (na API/CLI, eles são chamados de Endpoints de leitura).
- Para clusters do Valkey ou do Redis OSS (modo cluster habilitado), use o Endpoint de configuração do cluster para todas as operações com suporte a comandos do modo cluster habilitado. Você deve usar um cliente que ofereça suporte ao cluster do Valkey ou ao cluster do Redis OSS no Redis OSS 3.2 e superior. Você ainda pode ler de endpoints de nó individuais (na API/CLI, eles se chamam Endpoints de leitura).

As seções a seguir orientam pelo processo de localização dos endpoints necessários para o mecanismo que você está executando.

Quais endpoints usar com o Memcached.

Para o cache com tecnologia sem servidor do ElastiCache para o Memcached, basta adquirir o DNS do endpoint do cluster e a porta do console.

Na AWS CLI, use o comando `describe-serverless-caches` para adquirir as informações do endpoint.

Linux

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

Windows

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

O resultado da operação acima deve ser semelhante a este (formato 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"
    }
  ]
}
```

Para um cluster do Memcached com base na instância, se usar a descoberta automática, você poderá usar o endpoint de configuração do cluster para configurar o cliente do Memcached. Isso significa que você deve usar um cliente que ofereça suporte para Descoberta automática.

Se você não usar a Descoberta automática, deverá configurar seu cliente para usar os endpoints de nó individuais para leituras e gravações. Você também deve acompanhá-los à medida que adiciona e remove nós.

Localização de endpoints de um cluster do Valkey ou do Redis OSS (modo cluster desabilitado) (console)

Se um cluster do Valkey ou Redis OSS (modo cluster desabilitado) tiver apenas um nó, o endpoint desse nó será usado para tanto para leituras quanto para gravações. Se um cluster do Valkey ou Redis OSS (modo cluster desabilitado) tiver vários nós, há três tipos de endpoints: o endpoint primário, o endpoint leitor e os endpoints de nó.

O endpoint primário é um nome DNS que sempre é resolvido para o nó primário no cluster. O endpoint primário é imune às alterações no seu cluster, como promover uma réplica de leitura para a função primária. Para atividades de gravação, recomendamos que suas aplicações se conectem ao endpoint primário.

Um endpoint de leitor divide uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura em um cluster do ElastiCache para Redis OSS. Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar várias réplicas de leitura do seu cluster do ElastiCache para Redis OSS em diferentes zonas de disponibilidade da AWS (AZ) para garantir a alta disponibilidade dos endpoints leitores.

Note

Um endpoint leitor não é um balanceador de carga. É um registro DNS que vai resolver para um endereço IP de um dos nós de réplica em uma forma round robin.

Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó no seu aplicativo.

Para descobrir endpoints de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.

A tela de clusters aparecerá com uma lista de clusters do Valkey ou Redis OSS (modo cluster desabilitado) e Valkey ou Redis OSS (modo cluster habilitado).

3.
- Para encontrar os endpoints Primário e/ou Leitor do cluster, escolha o nome do cluster (e não o botão à esquerda).

▼ Cluster details			
Cluster name	Description	Node type	Status
		cache.r6g.large	Available
Engine	Engine version	Global datastore	Global datastore role
Redis OSS	6.0.5	-	-
Update status	Cluster mode	Shards	Number of nodes
Update available	Off	1	3
Data tiering	Multi-AZ	Auto-failover	Encryption in transit
Disabled	Enabled	Enabled	Disabled
Encryption at rest	Parameter group	Outpost ARN	Configuration endpoint
Disabled	default.redis6.x	-	-
Primary endpoint	Reader endpoint	ARN	
-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379		

Endpoints primários e leitores para um cluster do Valkey ou Redis OSS (modo cluster desabilitado)

Se houver apenas um nó no cluster, não haverá um endpoint primário, e você poderá continuar na próxima etapa.

4.
- Se o cluster do Valkey ou Redis OSS (modo cluster desabilitado) tiver nós de réplica, você poderá encontrar os endpoints dos nós de réplica do cluster escolhendo o nome do cluster e escolhendo a guia Nós.

A tela de nós aparece com cada nó do cluster, primário e réplicas, listados com seu respectivo endpoint.

☐	Node Name	▲	Status	Current Role	Port	Endpoint
☐	test-no-001		available	primary	6379	-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com
☐	test-no-002		available	replica	6379	-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com
☐	test-no-003		available	replica	6379	-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com

Endpoints de nó para um cluster do Valkey ou Redis OSS (modo cluster desabilitado)

5. Como copiar um endpoint para a área de transferência:
 - a. Um endpoint por vez, localize o endpoint que você deseja copiar.
 - b. Escolha o ícone de cópia diretamente na frente do endpoint.

O endpoint agora é copiado para a área de transferência. Para obter informações sobre como usar o endpoint para se conectar a um nó, consulte [Conexão a nós do Memcached](#).

Um endpoint primário do Valkey ou Redis OSS (modo cluster desabilitado) se assemelha ao seguinte. Há uma diferença, dependendo de a criptografia em trânsito estar ou não habilitada.

Criptografia em trânsito não habilitada

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

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

Criptografia em trânsito habilitada

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

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

Localização de endpoints para um cluster do Valkey ou do Redis OSS (modo cluster habilitado) (console)

O cluster do Valkey ou Redis OSS (modo cluster habilitado) possui um único endpoint de configuração. Ao se conectar ao endpoint de configuração, a aplicação pode descobrir os endpoints primários e os endpoints de leitura para cada fragmento no cluster.

Para encontrar o endpoint de um cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.

- A tela de clusters será exibida com uma lista de clusters. Escolha o cluster ao qual você deseja se conectar.
3. Para encontrar o endpoint de configuração do cluster, escolha o nome do cluster (não o botão de opção).
 4. O endpoint de configuração é exibido em detalhes do cluster. Para copiá-lo, selecione o ícone copiar à esquerda do endpoint.

Localização de endpoints de um cluster (console) (Memcached)

Todos os endpoints Memcached são endpoints de leitura/gravação. Para se conectar a nós em um cluster Memcached, seu aplicativo pode usar os endpoints para cada nó ou o endpoint de configuração do cluster, juntamente com a Descoberta automática. Para usar a Descoberta automática, você deve usar um cliente que ofereça suporte para Descoberta automática.

Ao usar a Descoberta automática, seu aplicativo cliente se conecta ao seu cluster Memcached usando o endpoint de configuração. À medida que você escalar seu cluster adicionando ou removendo nós, seu aplicativo "conhecerá" automaticamente todos os nós do cluster e poderá se conectar a qualquer um deles. Sem a Descoberta automática, seu aplicativo teria que fazer isso, ou você precisaria atualizar manualmente os endpoints no seu aplicativo sempre que adicionasse ou removesse um nó.

Para copiar um endpoint, escolha o ícone de cópia diretamente na frente do endereço do endpoint. Para obter informações sobre como usar o endpoint para se conectar a um nó, consulte [Conexão a nós do Memcached](#).

Endpoints de configuração e de nó são muitos semelhantes. As diferenças estão realçadas com negrito a seguir.

```
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

Se você optar por criar um CNAME para o seu endpoint de configuração Memcached, para que o seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir `.cfg.` nesse CNAME.

Localizar endpoints (AWS CLI)

Para o Memcached, você pode usar a AWS CLI do Amazon ElastiCache para descobrir os endpoints para nós e clusters.

Para o Redis OSS, você pode usar a AWS CLI do Amazon ElastiCache para descobrir os endpoints para nós, clusters e também grupos de replicação.

Tópicos

- [Encontrar endpoints para nós e clusters \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(AWS CLI\)](#)

Encontrar endpoints para nós e clusters (AWS CLI)

Você pode usar a AWS CLI para descobrir os endpoints para um cluster e seus nós com o comando `describe-cache-clusters`. Para clusters do Valkey ou Redis OSS, o comando retorna o endpoint do cluster. Para clusters Memcached, o comando retorna o endpoint de configuração. Se você incluir o parâmetro opcional `--show-cache-node-info`, o comando também retornará os endpoints dos nós individuais no cluster.

Example

O comando a seguir recupera os endpoints de configuração (`ConfigurationEndpoint`) e os endpoints de nó individuais (`Endpoint`) para o cluster Memcached `mycluster`.

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws elasticache describe-cache-clusters ^
  --cache-cluster-id mycluster ^
  --show-cache-node-info
```

O resultado da operação acima deve ser semelhante a este (formato JSON).

```
{
```

```
"CacheClusters": [
{
  "Engine": "memcached",
  "CacheNodes": [
    {
      "CacheNodeId": "0001",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    },
    {
      "CacheNodeId": "0002",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    },
    {
      "CacheNodeId": "0003",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    }
  ],
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.memcached1.4",
    "ParameterApplyStatus": "in-sync"
  },
  "CacheClusterId": "mycluster",
```



```

    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "available",
    "NumCacheNodes": 3,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
    "CacheNodeType": "cache.m4.large",
    "DataTiering": "disabled"
}
]
}

```

Important

Se você optar por criar um CNAME para o seu endpoint de configuração Memcached, para que o seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir `.cfg.` nesse CNAME. Por exemplo, `mycluster.cfg.local` no seu arquivo `php.ini` para o parâmetro `session.save_path`.

Example

Para Valkey e Redis OSS, o comando a seguir recupera as informações do cluster para o cluster de nó único mycluster.

Important

O parâmetro `--cache-cluster-id` pode ser usado com um ID de cluster do Valkey ou Redis OSS (modo cluster desabilitado) de nó único ou IDs de nó específicos em grupos de replicação. O `--cache-cluster-id` de um grupo de replicação é um valor de 4

dígitos, como 0001. Se `--cache-cluster-id` for o ID de um cluster (nó) em um grupo de replicação, o `replication-group-id` será incluído na saída.

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id redis-cluster ^  
  --show-cache-node-info
```

O resultado da operação acima deve ser semelhante a este (formato JSON).

```
{  
  "CacheClusters": [  
    {  
      "CacheClusterStatus": "available",  
      "SecurityGroups": [  
        {  
          "SecurityGroupId": "sg-77186e0d",  
          "Status": "active"  
        }  
      ],  
      "CacheNodes": [  
        {  
          "CustomerAvailabilityZone": "us-east-1b",  
          "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",  
          "CacheNodeStatus": "available",  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Address": "redis-cluster.amazonaws.com",  
            "Port": 6379  
          },  
          "ParameterGroupStatus": "in-sync"  
        }  
      ],  
      "AtRestEncryptionEnabled": false,  
    }  
  ]  
}
```

```

    "CacheClusterId": "redis-cluster",
    "TransitEncryptionEnabled": false,
    "CacheParameterGroup": {
        "ParameterApplyStatus": "in-sync",
        "CacheNodeIdsToReboot": [],
        "CacheParameterGroupName": "default.redis3.2"
    },
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-east-1b",
    "AutoMinorVersionUpgrade": true,
    "Engine": "redis",
    "AuthTokenEnabled": false,
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",
    "CacheSecurityGroups": [],
    "CacheSubnetGroupName": "default",
    "CacheNodeType": "cache.t2.small",
    "DataTiering": "disabled",
    "EngineVersion": "3.2.10",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
  }
]
}

```

Para obter mais informações, consulte o tópico [describe-cache-clusters](#).

Localização de endpoints para grupos de replicação do Valkey ou Redis OSS (AWS CLI)

Você pode usar a AWS CLI para descobrir os endpoints para um grupo de replicação e seus clusters com o comando `describe-replication-groups`. O comando retorna o endpoint primário do grupo de replicação e uma lista de todos os clusters (nós) no grupo de replicação com seus endpoints, junto com o endpoint de leitor.

A seguinte operação recupera o endpoint primário e o endpoint de leitor para o grupo de replicação `myreplgroup`. Use o endpoint primário para todas as operações de gravação.

```

aws elasticache describe-replication-groups \
  --replication-group-id myreplgroup

```

Para Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id myreplgroup
```

O resultado dessa operação deve ser semelhante a este (formato JSON).

```
{  
  "ReplicationGroups": [  
    {  
      "Status": "available",  
      "Description": "test",  
      "NodeGroups": [  
        {  
          "Status": "available",  
          "NodeGroupMembers": [  
            {  
              "CurrentRole": "primary",  
              "PreferredAvailabilityZone": "us-west-2a",  
              "CacheNodeId": "0001",  
              "ReadEndpoint": {  
                "Port": 6379,  
                "Address": "myreplgroup-001.amazonaws.com"  
              },  
              "CacheClusterId": "myreplgroup-001"  
            },  
            {  
              "CurrentRole": "replica",  
              "PreferredAvailabilityZone": "us-west-2b",  
              "CacheNodeId": "0001",  
              "ReadEndpoint": {  
                "Port": 6379,  
                "Address": "myreplgroup-002.amazonaws.com"  
              },  
              "CacheClusterId": "myreplgroup-002"  
            },  
            {  
              "CurrentRole": "replica",  
              "PreferredAvailabilityZone": "us-west-2c",  
              "CacheNodeId": "0001",  
              "ReadEndpoint": {  
                "Port": 6379,  
                "Address": "myreplgroup-003.amazonaws.com"  
              },  
              "CacheClusterId": "myreplgroup-003"  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```
        }
      ],
      "NodeGroupId": "0001",
      "PrimaryEndpoint": {
        "Port": 6379,
        "Address": "myreplgroup.amazonaws.com"
      },
      "ReaderEndpoint": {
        "Port": 6379,
        "Address": "myreplgroup-ro.amazonaws.com"
      }
    }
  ],
  "ReplicationGroupId": "myreplgroup",
  "AutomaticFailover": "enabled",
  "SnapshottingClusterId": "myreplgroup-002",
  "MemberClusters": [
    "myreplgroup-001",
    "myreplgroup-002",
    "myreplgroup-003"
  ],
  "PendingModifiedValues": {}
}
]
```

Para obter mais informações, consulte [describe-replication-groups](#) na Referência de comandos da AWS CLI.

Localização de endpoints (API do ElastiCache)

Para o Memcached, você pode usar a API do Amazon ElastiCache para descobrir os endpoints para nós e clusters.

Para o Redis OSS, você pode usar a API do Amazon ElastiCache para descobrir os endpoints para nós, clusters e também grupos de replicação.

Tópicos

- [Localização de endpoints para nós e clusters \(API do ElastiCache\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(API do ElastiCache\)](#)

Localização de endpoints para nós e clusters (API do ElastiCache)

Você pode usar a API do ElastiCache para descobrir os endpoints para um cluster e seus nós com a ação `DescribeCacheClusters`. Para clusters do Valkey ou Redis OSS, o comando retorna o endpoint do cluster. Para clusters Memcached, o comando retorna o endpoint de configuração. Se você incluir o parâmetro opcional `ShowCacheNodeInfo`, a ação também retornará os endpoints dos nós individuais no cluster.

Example

Para o Memcached, o comando a seguir recupera os endpoints de configuração (`ConfigurationEndpoint`) e os endpoints de nó individuais (`Endpoint`) para o cluster do 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

Se você optar por criar um CNAME para o seu endpoint de configuração Memcached, para que o seu cliente de descoberta automática reconheça o CNAME como um

endpoint de configuração, você deverá incluir `.cfg` nesse CNAME. Por exemplo, `mycluster.cfg.local` no seu arquivo `php.ini` para o parâmetro `session.save_path`.

Localização de endpoints para grupos de replicação do Valkey ou Redis OSS (API do ElastiCache)

Você pode usar a API do ElastiCache para descobrir os endpoints para um grupo de replicação e seus cluster com a ação `DescribeReplicationGroups`. A ação retorna o endpoint primário do grupo de replicação e uma lista de todos os clusters no grupo de replicação com seus endpoints, junto com o endpoint de leitor.

A operação a seguir recupera o endpoint primário (`PrimaryEndpoint`), o endpoints de leitor (`ReaderEndpoint`) e os endpoints de nós individuais (`ReadEndpoint`) para o grupo de replicação `myreplgroup`. Use o endpoint primário para todas as operações de gravação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=myreplgroup  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeReplicationGroups](#).

Operação com fragmentos no ElastiCache

Um fragmento (API/CLI: grupo de nós) é uma coleção de um a seis nós do ElastiCache para Valkey ou Redis OSS. Um cluster do Valkey ou Redis OSS (modo cluster desabilitado) nunca terá mais de um fragmento. Com os fragmentos, você pode dividir grandes bancos de dados em partes menores, mais rápidas e mais fáceis de gerenciar, chamadas de fragmentos de dados. Isso pode aumentar a eficiência do banco de dados distribuindo as operações em várias seções separadas. O uso de fragmentos pode oferecer muitos benefícios, incluindo melhor desempenho, escalabilidade e eficiência de custos.

É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas. Os dados do cluster são

particionados entre todos os fragmentos do cluster. Se houver mais de um nó em um fragmento, este implementará a replicação com um nó sendo o nó primário de leitura/gravação e os outros nós como nós de réplica somente leitura.

O limite de nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo for Valkey 7.2 e posteriores ou Redis OSS 5.0.6 a 7.1. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

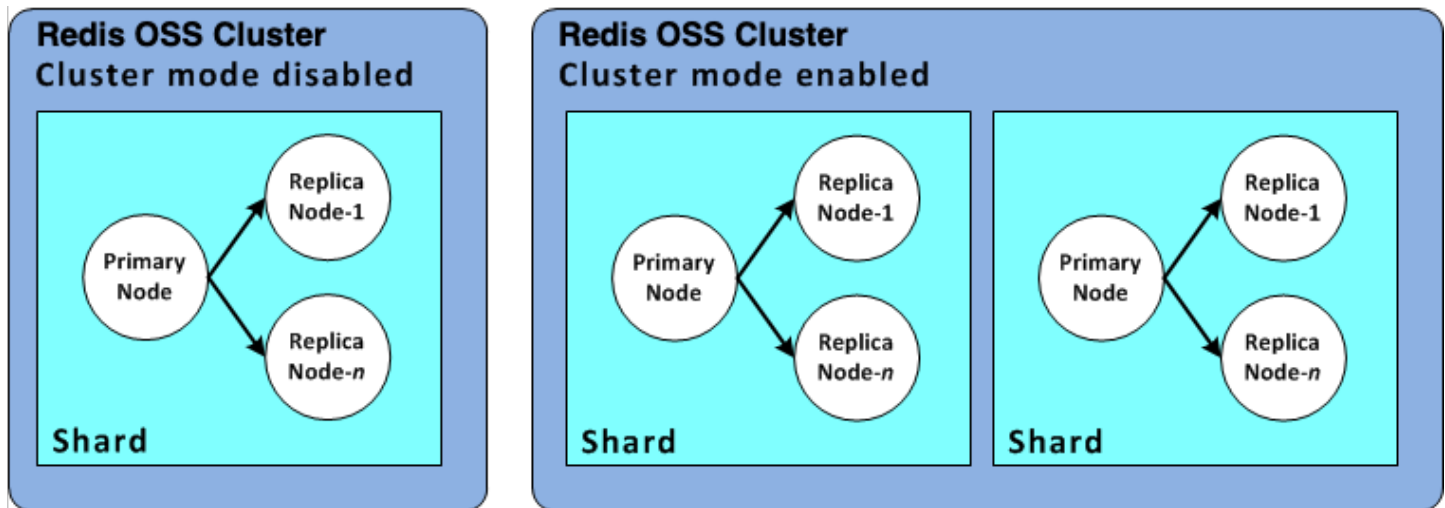
Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites do serviço da AWS](#) e escolha o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Ao criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) usando o console do ElastiCache, você especifica o número de fragmentos no cluster e o número de nós nos fragmentos. Para obter mais informações, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#). Se você usar a API do ElastiCache ou a AWS CLI para criar um cluster (chamado de grupo de replicação na API/CLI), poderá configurar o número de nós em um fragmento (API/CLI: grupo de nós) de forma independente. Para obter mais informações, consulte:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Cada nó em um fragmento tem as mesmas especificações de computação, armazenamento e memória. A API do ElastiCache permite que você controle atributos no âmbito do fragmento, como o número de nós, as configurações de segurança e as janelas de manutenção do sistema.



Configurações de fragmento do Valkey ou Redis OSS

Para obter mais informações, consulte [Refragmentação offline para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#) e [Refragmentação online para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#).

Localização de um ID de fragmento

Você pode encontrar o ID de um fragmento usando o Console de gerenciamento da AWS, a AWS CLI ou a API do ElastiCache.

Utilização do Console de gerenciamento da AWS

Tópicos

- [Para Valkey ou Redis OSS \(modo cluster desabilitado\)](#)
- [Para Valkey ou Redis OSS \(modo cluster habilitado\)](#)

Para Valkey ou Redis OSS (modo cluster desabilitado)

Os IDs de fragmento do grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) são sempre 0001.

Para Valkey ou Redis OSS (modo cluster habilitado)

O procedimento a seguir usa o Console de gerenciamento da AWS para localizar o ID de um fragmento do Valkey ou Redis OSS (modo cluster habilitado).

Para localizar o ID do fragmento em um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS e o nome do grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) para o qual deseja encontrar os IDs do fragmento.
3. Na coluna Shard Name (Nome do fragmento), o ID do fragmento tem os últimos quatro dígitos do nome do fragmento.

Como usar o AWS CLI

Para encontrar os IDs de fragmento (grupo de nós) para grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado) ou do Valkey ou Redis OSS (modo cluster habilitado), use a operação da AWS CLI `describe-replication-groups` com o parâmetro opcional a seguir.

- **--replication-group-id**—um parâmetro opcional que, quando usado, limita a saída aos detalhes do grupo de replicação especificado. Caso esse parâmetro seja omitido, os detalhes de até 100 grupos de replicação são retornados.

Example

Esse comando retorna os detalhes do `sample-repl-group`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-replication-groups \
  --replication-group-id sample-repl-group
```

Para Windows:

```
aws elasticache describe-replication-groups ^
  --replication-group-id sample-repl-group
```

A saída desse comando é semelhante a esta. Os IDs do estilhaço (grupo de nós) estão *realçados* aqui para facilitar a localização deles.

```
{
```

```

"ReplicationGroups": [
  {
    "Status": "available",
    "Description": "2 shards, 2 nodes (1 + 1 replica)",
    "NodeGroups": [
      {
        "Status": "available",
        "Slots": "0-8191",
        "NodeGroupId": "0001",
        "NodeGroupMembers": [
          {
            "PreferredAvailabilityZone": "us-west-2c",
            "CacheNodeId": "0001",
            "CacheClusterId": "sample-repl-group-0001-001"
          },
          {
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheNodeId": "0001",
            "CacheClusterId": "sample-repl-group-0001-002"
          }
        ]
      },
      {
        "Status": "available",
        "Slots": "8192-16383",
        "NodeGroupId": "0002",
        "NodeGroupMembers": [
          {
            "PreferredAvailabilityZone": "us-west-2b",
            "CacheNodeId": "0001",
            "CacheClusterId": "sample-repl-group-0002-001"
          },
          {
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheNodeId": "0001",
            "CacheClusterId": "sample-repl-group-0002-002"
          }
        ]
      }
    ],
    "ConfigurationEndpoint": {
      "Port": 6379,
      "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
    }
  }
]

```

```

    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "sample-repl-group",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "13:00-14:00",
    "MemberClusters": [
        "sample-repl-group-0001-001",
        "sample-repl-group-0001-002",
        "sample-repl-group-0002-001",
        "sample-repl-group-0002-002"
    ],
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
    "PendingModifiedValues": {}
  }
]
}

```

Uso da API do ElastiCache

Para encontrar os IDs de fragmento (grupo de nós) para grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado) ou do Valkey ou Redis OSS (modo cluster habilitado), use a operação da AWS CLI `describe-replication-groups` com o parâmetro opcional a seguir.

- **ReplicationGroupId**—um parâmetro opcional que, quando usado, limita a saída aos detalhes do grupo de replicação especificado. Caso esse parâmetro seja omitido, os detalhes de até **xxx** grupos de replicação são retornados.

Example

Esse comando retorna os detalhes do `sample-repl-group`.

Para Linux, macOS ou Unix:

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroup
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z

```

```
&X-Amz-Credential=<credential>
```

Comparação de clusters Valkey, Memcached e Redis OSS baseados em nós

A Amazon ElastiCache oferece suporte aos mecanismos de cache Valkey, Memcached e Redis OSS. Cada mecanismo oferece algumas vantagens. Use as informações contidas neste tópico para ajudá-lo a escolher o mecanismo e a versão que melhor atendam aos seus requisitos.

Important

Depois de criar um cluster de cache baseado em nós ou grupo de replicação, você pode atualizar para uma versão de mecanismo mais recente, mas não pode fazer downgrade para uma versão de mecanismo mais antiga. Se quiser usar uma versão de mecanismo mais antiga, deverá excluir o cluster de cache baseado em nós ou grupo de replicação existente e criá-lo novamente com a versão anterior do mecanismo.

Na superfície, os mecanismos parecem semelhantes. Cada um deles é um armazenamento de chaves/valores na memória. No entanto, na prática, existem diferenças significativas.

Escolha o Memcached se o seguinte se aplicar a você:

- Você precisa do modelo mais simples possível.
- Você precisa executar nós grandes com vários cores ou threads.
- Você precisa da capacidade de expandir e reduzir, adicionar e remover nós à medida que a demanda no seu sistema aumenta e diminui.
- Você precisa armazenar objetos em cache.

Escolha Valkey ou Redis OSS com ElastiCache se o seguinte se aplicar a você:

- ElastiCache versão 7.2 para Valkey ou versão 7.0 (aprimorada) para Redis OSS

Você quer usar [Funções](#), [Pub/Sub fragmentado](#) ou [Melhorias de ACL](#). Para obter mais informações, consulte [Redis OSS versão 7.0 \(aprimorada\)](#).

- ElastiCache versão 6.2 (aprimorada) para Redis OSS

Você deseja ter a capacidade de classificar dados em níveis entre memória e SSD usando o tipo de nó r6gd. Para mais informações, consulte [Classificação de dados em níveis](#).

- ElastiCache versão 6.0 (aprimorada) para Redis OSS

Você deseja autenticar usuários com o controle de acesso baseado em função.

Para obter mais informações, consulte [Redis OSS versão 6.0 \(aprimorada\)](#).

- ElastiCache versão 5.0.0 (aprimorada) para Redis OSS

Você quer usar [Fluxos Redis OSS](#), uma estrutura de dados de log que permite aos produtores anexarem novos itens em tempo real e também permite que os consumidores consumam mensagens de forma bloqueante ou não bloqueante.

Para obter mais informações, consulte [Redis OSS versão 5.0.0 \(aprimorada\)](#).

- ElastiCache versão 4.0.10 (aprimorada) para Redis OSS

Oferece suporte à criptografia e à adição ou remoção de fragmentos do cluster do Valkey ou do Redis OSS (modo cluster habilitado) de modo dinâmico.

Para obter mais informações, consulte [Redis OSS versão 4.0.10 \(aprimorada\)](#).

As versões a seguir estão obsoletas ou chegaram ou estão prestes a chegar ao fim da vida útil.

- ElastiCache versão 3.2.10 (aprimorada) para Redis OSS

Oferece suporte à adição ou remoção de fragmentos do cluster do Valkey ou do Redis OSS (modo cluster habilitado) de modo dinâmico.

 Important

Atualmente, o ElastiCache 3.2.10 para Redis OSS não oferece suporte à criptografia.

Para saber mais, consulte:

- [Redis OSS versão 3.2.10 \(aprimorado\)](#)
- Refragmentação online de melhores práticas para o Redis OSS. Para obter mais informações, consulte o seguinte:

- [Práticas recomendadas: reestilhecimento online](#)
- [Refragmentação e rebalanceamento de fragmentos online para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#)
- Para obter mais informações sobre como escalonar clusters Redis OSS, consulte [Escalaabilidade](#).
- ElastiCache versão 3.2.6 (aprimorada) para Redis OSS

Se você precisar da funcionalidade de versões anteriores do Redis OSS, bem como dos atributos a seguir, escolha o 3.2.6:

- Criptografia em trânsito. Para obter mais informações, consulte [Amazon ElastiCache for Redis OSS In-Transit Encryption](#).
- Criptografia em repouso. Para obter mais informações, consulte [Amazon ElastiCache for Redis OSS At-Rest Encryption](#).
- ElastiCache (Modo de cluster ativado) versão 3.2.4 para Redis OSS

Se precisar da funcionalidade do 2.8.x, bem como dos atributos a seguir, escolha o 3.2.4 (modo de cluster):

- Você precisa dividir seus dados entre dois a 500 grupos de nós (somente no modo de cluster).
- Você precisa de indexação geoespacial (modo de cluster ou modo sem cluster).
- Você não precisa comportar vários bancos de dados.
- ElastiCache (modo sem cluster) 2.8.x e 3.2.4 (aprimorado) para Redis OSS

Se o seguinte se aplicar a você, escolha o 2.8.x ou 3.2.4 (modo sem cluster):

- Você precisa de tipos de dados complexos, como strings, hashes, listas, conjuntos, conjuntos classificados e bitmaps.
- Você precisa classificar ou organizar conjuntos de dados na memória.
- Você precisa de persistência do seu armazenamento de chaves.
- Você precisa replicar seus dados do primário para uma ou mais réplicas de leitura para aplicativos de leitura intensa.
- Você precisa de failover automático se o primário falhar.
- Você precisa publicar e assinar (pub/sub) recursos, para informar os clientes sobre eventos no servidor.

- Você precisa de recursos de backup e restauração para clusters baseado em nós, bem como para caches com tecnologia sem servidor.
- Você precisa oferecer suporte para vários bancos de dados.

Resumo da comparação do Memcached, Valkey ou Redis OSS (modo cluster desabilitado) e Valkey ou Redis OSS (modo cluster habilitado)

	Memcached	Valkey ou Redis OSS (modo cluster desabilitado)	Valkey ou Redis OSS (modo cluster habilitado)
Versões do motor+	1.4.5 e versões posteriores	4.0.10 e posterior	4.0.10 e posterior
Tipos de dados	Simples	2.8.x – Complexo* Complexo	3.2.x e posterior - Complexo
Particionamento de dados	Sim	Não	Sim
O cluster é modificável	Sim	Sim	3.2.10 e posterior - Limitado
Reestilhecimento online	Não	Não	3.2.10 e posterior
Criptografia	em trânsito 1.6.12 e versões posteriores	4.0.10 e posterior	4.0.10 e posterior
Hierarquização de dados	Não	6.2 e posteriores	6.2 e posteriores
Certificações de conformidade			
Certificação de compatibilidade			
FedRAMP	Sim, 1.6.12 e posterior	4.0.10 e posterior	4.0.10 e posterior
HIPAA	Sim, 1.6.12 e posterior	4.0.10 e posterior	4.0.10 e posterior
PCI DSS	Sim	4.0.10 e posterior	4.0.10 e posterior

	Memcached	Valkey ou Redis OSS (modo cluster desabilitado)	Valkey ou Redis OSS (modo cluster habilitado)
Multi-threaded	Sim	Não	Não
Atualização do tipo de nó	Não	Sim	Sim
Atualização do mecanismo	Sim	Sim	Sim
Alta disponibilidade (replicação)	Não	Sim	Sim
Failover automático	Não	Opcional	Obrigatório
Recursos de publicação/assinatura (pub/sub)	Não	Sim	Sim
Conjuntos classificados	Não	Sim	Sim
Backup e restauração	Somente para caches sem servidor, não aplicável a clusters baseados em nós	Sim	Sim
Indexação geoespacial	Não	4.0.10 e posterior	Sim

Observações:

string, objetos (como bancos de dados)

* strings, conjuntos, conjuntos classificados, listas, hashes, bitmaps, hiperloglog

strings, conjuntos, conjuntos classificados, listas, hashes, bitmaps, hyperloglog, índices geoespaciais

Memcached

Valkey ou Redis OSS
(modo cluster desabilitado)Valkey ou Redis
OSS (modo cluster
habilitado)

+ Exclui versões que estão obsoletas, que atingiram ou estão prestes a atingir o fim da vida útil.

Depois de escolher o mecanismo do seu cluster, recomendamos que você use a versão mais recente desse mecanismo. Para obter mais informações, consulte [Tipos de nó compatíveis](#).

Migração on-line para o Valkey ou Redis OSS

Ao usar a migração on-line, você pode migrar seus dados do Valkey ou Redis OSS do seu código-fonte aberto e hospedagem própria no Amazon EC2 para o Amazon ElastiCache.

Isso se refere à migração de uma instância auto-hospedada para o serviço do ElastiCache. Para obter informações sobre a atualização do Redis OSS para o Valkey no ElastiCache, consulte [Atualização das versões do mecanismo, incluindo atualizações de mecanismos cruzados](#).

Note

A migração on-line não é compatível com caches ou clusters sem servidor do ElastiCache em execução no tipo de nó r6gd.

Visão geral

Migrar seus dados do Valkey ou Redis OSS de código-fonte aberto em execução no Amazon EC2 para o Amazon ElastiCache exige uma implantação do Amazon ElastiCache existente ou recém-criada. A implantação deve ter uma configuração pronta para migração. Ela também deve estar alinhada à configuração desejada, incluindo atributos, como tipo de instância, número de fragmentos e número de réplicas.

A migração on-line foi projetada para migrar dados do Valkey ou Redis OSS de código-fonte aberto e hospedagem própria no Amazon EC2 para o ElastiCache, e não para mover dados entre clusters do ElastiCache.

⚠ Important

É altamente recomendável ler integralmente as seções a seguir antes de iniciar o processo de migração online.

A migração começa quando você chama o comando da `StartMigration` ou a operação de API AWS CLI. Ao migrar clusters do Valkey ou Redis OSS de modo cluster desabilitado, o processo de migração torna o nó primário do cluster do ElastiCache Valkey ou Redis OSS uma réplica do Valkey ou Redis OSS primário de origem. Ao migrar clusters do Valkey ou Redis OSS de modo cluster habilitado, o processo de migração torna o nó primário de cada fragmento do ElastiCache uma réplica do fragmento correspondente do cluster de origem que tem os mesmos slots.

Depois que as alterações no lado do cliente estiverem prontas, chame a operação da API `CompleteMigration`. Essa operação de API promove a implantação do ElastiCache para sua implantação primária do Valkey ou Redis OSS com os nós principal e de réplica (conforme aplicável). Agora é possível redirecionar sua aplicação cliente para começar a gravar dados no ElastiCache. Durante a migração, é possível conferir o status da replicação executando o comando [valkey-cli INFO](#) nos nós do Valkey e nos nós primários do ElastiCache.

Etapas da migração

Os seguintes tópicos descrevem o processo de migração dos seus dados:

- [Preparação da origem e destino para migração](#)
- [Testar a migração de dados](#)
- [Início da migração](#)
- [Verificação do andamento do processo de migração de dados](#)
- [Conclusão da migração de dados](#)

Preparação da origem e destino para migração

Com essas etapas, você pode se preparar para migrar dados de uma origem de hospedagem própria do Valkey ou Redis no EC2 para o ElastiCache, ou de um cluster do Redis OSS para um cluster do ElastiCache Valkey.

Isso se refere à migração de uma instância auto-hospedada para o serviço do ElastiCache. Para obter informações sobre a atualização do Redis OSS para o Valkey no ElastiCache, consulte [Atualização das versões do mecanismo, incluindo atualizações de mecanismos cruzados](#).

É necessário garantir que todos os quatro pré-requisitos mencionados a seguir sejam cumpridos antes de iniciar a migração do console do ElastiCache, da API ou da CLI da AWS.

Como preparar os nós de origem e destino do Valkey ou Redis OSS para migração

1. Identifique a implantação do ElastiCache de destino e verifique se é possível migrar dados para ela.

Uma implantação do ElastiCache existente ou recém-criada deve atender aos seguintes requisitos para migração:

- Ele está usando o Valkey ou Redis OSS 5.0.6 ou posterior.
 - A criptografia em trânsito não está habilitada.
 - Ele tem o multi-AZ habilitado.
 - Ele tem memória suficiente disponível para comportar os dados do cluster do Valkey ou Redis OSS. Para configurar as definições de memória reservada corretas, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).
 - Para o modo cluster desabilitado, você pode migrar diretamente do Valkey ou Redis OSS versões 2.8.21 em diante para o Valkey ou Redis OSS versões 5.0.6 em diante se estiver usando a CLI ou o Valkey ou Redis OSS versões 5.0.6 em diante usando a CLI ou o console. Para o modo cluster habilitado, você pode migrar diretamente de qualquer versão do Valkey ou Redis OSS de modo cluster habilitado para o Redis OSS versão 5.0.6 em diante se estiver usando a CLI ou o Redis OSS versão 5.0.6 em diante usando a CLI ou o console.
 - O número de fragmentos na origem e no destino coincide.
 - Ele não faz parte de um datastore global.
 - Ele tem a hierarquização de dados desabilitada.
2. Certifique-se de que as configurações do Valkey ou Redis OSS de código aberto e a implantação do ElastiCache sejam compatíveis.

No mínimo, todos os itens a seguir na implantação de destino do ElastiCache devem ser compatíveis com sua configuração do Valkey ou Redis OSS para replicação:

- O cluster não deve ter o AUTH habilitado.

- O config `protected-mode` deve estar definido como `no`.
 - Se você tiver a configuração `bind` no config do Valkey ou Redis OSS, ela deve ser atualizada para permitir solicitações de nós do ElastiCache.
 - O número de bancos de dados lógicos deve ser o mesmo no nó do ElastiCache e no cluster do Valkey ou Redis OSS. Esse valor é definido usando `databases` no config do Valkey ou Redis OSS.
 - Os comandos do Valkey ou Redis OSS que executam a modificação de dados não devem ser renomeados para permitir que a replicação dos dados seja bem-sucedida. Por exemplo, `sync`, `psync`, `info`, `config`, `command` e `cluster`.
 - Para replicar os dados do cluster do Valkey ou Redis OSS para o ElastiCache, verifique se há CPU e memória suficientes para lidar com essa carga adicional. Essa carga vem do arquivo RDB criado pelo cluster do Valkey ou Redis OSS e transferido pela rede para o nó do ElastiCache.
 - Todas as instâncias do Valkey ou Redis OSS no cluster de origem devem estar em execução na mesma porta.
3. Verifique se as instâncias do EC2 podem se conectar ao ElastiCache fazendo o seguinte:
- Garanta que cada endereço IP da instância seja privado.
 - Atribua ou crie a implantação do ElastiCache na mesma nuvem privada virtual (VPC) do Valkey ou Redis OSS na instância (recomendado).
 - Se as VPCs forem diferentes, configure o emparelhamento de VPCs para permitir o acesso entre os nós. Para obter mais informações sobre o emparelhamento de VPCs, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#).
 - O grupo de segurança anexado às instâncias do Valkey ou Redis OSS deve permitir o tráfego de entrada de nós do ElastiCache.
4. Verifique se a aplicação pode direcionar o tráfego para nós do ElastiCache após a conclusão da migração de dados. Para obter mais informações, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#).

Testar a migração de dados

Depois que todos os pré-requisitos forem atendidos, será possível iniciar a migração de dados usando o Console de gerenciamento da AWS, a API do ElastiCache ou a AWS CLI. O exemplo a seguir mostra como usar a CLI.

Teste a migração chamando o comando `test-migration` com os seguintes parâmetros:

- `--replication-group-id`: o ID do grupo de replicação para o qual os dados devem ser migrados.
- `--customer-node-endpoint-list`: lista de endpoints dos quais os dados devem ser migrados. A lista deve ter somente um elemento.

Veja a seguir um exemplo de como usar a CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

O ElastiCache validará a configuração da migração sem nenhuma migração real de dados.

Início da migração

Depois que todos os pré-requisitos forem preenchidos, será possível iniciar a migração de dados usando o Console de gerenciamento da AWS, a API do ElastiCache ou a AWS CLI. No modo de cluster habilitado, se a migração de slots for diferente, uma refragmentação será realizada antes da migração ao vivo. O exemplo a seguir mostra como usar a CLI.

Note

Recomendamos usar a API `TestMigration` para validar a configuração da migração. Mas isso é estritamente opcional.

Inicie a migração chamando o comando `start-migration` com os seguintes parâmetros:

- `--replication-group-id`: identificador do grupo de replicação de destino do ElastiCache.
- `--customer-node-endpoint-list`: uma lista de endpoints com endereços IP ou DNS e a porta em que o cluster de origem do Valkey ou Redis OSS está sendo executado. A lista só pode ter um elemento para o modo de cluster desabilitado e o modo de cluster habilitado. Se você habilitou a replicação encadeada, o endpoint pode apontar para uma réplica em vez de um nó primário no cluster do Valkey ou Redis OSS.

Veja a seguir um exemplo de como usar a CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Ao executar esse comando, o nó primário do ElastiCache (em cada fragmento) se configura para se tornar uma réplica da instância do Valkey ou Redis OSS (no fragmento correspondente que tem os mesmos slots no redis habilitado para cluster). O status do cluster do ElastiCache é alterado para migrando e os dados começam a ser migrados da instância do Valkey ou Redis OSS para o nó primário do ElastiCache. Dependendo do tamanho dos dados e da carga na instância do Valkey ou Redis OSS, a migração pode levar um tempo até ser concluída. É possível conferir o andamento da migração executando o comando [valkey-cli INFO](#) na instância do Valkey e no nó primário do ElastiCache.

Após a replicação bem-sucedida, todas as gravações nas instâncias do Valkey ou Redis OSS são propagadas para o cluster do ElastiCache. É possível usar os nós do ElastiCache para leituras. No entanto, não é possível gravar no cluster do ElastiCache. Se um nó primário do ElastiCache tiver outros nós de replica conectados a ele, esse nó de réplica continuará a replicar do nó primário do ElastiCache. Dessa maneira, todos os dados do cluster do Valkey ou Redis OSS serão replicados para todos os nós no cluster do ElastiCache.

Se um nó primário do ElastiCache não puder se tornar uma réplica da instância do Valkey ou Redis OSS, ele fará várias novas tentativas antes de finalmente se promover novamente a primário. O status do cluster do ElastiCache é, então, alterado para available (disponível) e é enviado um evento do grupo de replicação sobre a falha ao iniciar a migração. Para solucionar problemas como uma falha, verifique o seguinte:

- Observe o evento do grupo de replicação. Use qualquer informação específica do evento para corrigir a falha na migração.
- Se o evento não fornecer informações específicas, verifique se as diretrizes em foram seguidas [Preparação da origem e destino para migração](#).
- Assegure que a configuração de roteamento para a VPC e as sub-redes permitam tráfego entre os nós do ElastiCache e as instâncias do Valkey ou Redis OSS.
- Garanta que o grupo de segurança anexado às instâncias do Valkey ou Redis OSS permita tráfego de entrada de nós do ElastiCache.
- Verifique os logs do Valkey ou do Redis OSS para suas instâncias para obter mais informações sobre falhas específicas de replicação.

Verificação do andamento do processo de migração de dados

Após o início da migração de dados, é possível fazer o seguinte para monitorar o andamento:

- Verifique se o `master_link_status` do Valkey ou Redis OSS está up no comando `INFO` nos nós primários do ElastiCache. Também é possível encontrar essa informação no console do ElastiCache. Selecione o cluster e, em CloudWatch metrics (Métricas do CloudWatch), observe o Primary Link Health Status (Status de integridade do link principal). Depois que o valor atinge 1, os dados estão sincronizados.
- É possível conferir se a réplica do ElastiCache está no estado on-line executando o comando `INFO` nas instâncias do Valkey ou Redis OSS. Isso também fornece informações sobre o atraso da replicação.
- Verifique o buffer baixo de saída do cliente usando o comando [CLIENT LIST](#) nas instâncias do Valkey ou Redis OSS.

Após a conclusão da migração de dados, os dados estarão sincronizados com quaisquer novas gravações recebidas nos nós primários do cluster Valkey ou Redis OSS.

Conclusão da migração de dados

Quando estiver pronto para transferir ao cluster do ElastiCache, use o comando `complete-migration` da CLI com os seguintes parâmetros:

- `--replication-group-id`: o identificador do grupo de replicação.
- `--force`: um valor que força a interrupção da migração sem garantir que os dados estejam em sincronia.

Veja um exemplo a seguir.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

À medida que você executa esse comando, o nó primário do ElastiCache (em cada fragmento) interrompe a replicação da instância do Valkey ou Redis OSS e o promove a primário. Normalmente, essa promoção é concluída em alguns minutos. Para confirmar a promoção para primário, verifique o evento `Complete Migration successful for test-cluster`. Neste ponto, você pode direcionar sua aplicação para gravações e leituras do ElastiCache. O status do cluster do ElastiCache deve mudar de `migrating` (em migração) para `available` (disponível).

Se a promoção para primário falhar, o nó primário do ElastiCache continuará a replicar da instância do Valkey ou Redis OSS. O cluster do ElastiCache continuará no status migrating (em migração) e será enviada uma mensagem de evento do grupo de replicação sobre a falha. Para resolver essa falha, observe o seguinte:

- Verifique o evento do grupo de replicação. Use informações específicas do evento para corrigir a falha.
- Você pode receber uma mensagem de evento sobre os dados não estarem em sincronia. Se isso ocorrer, verifique se o primário do ElastiCache pode replicar da instância do Valkey ou Redis OSS e se os dois estão em sincronia. Se você ainda quiser interromper a migração, poderá executar o comando anterior com a opção `–force`.
- Você pode receber uma mensagem de evento se um dos nós do ElastiCache estiver passando por uma substituição. Você poderá tentar executar novamente a etapa de migração completa depois que a substituição for concluída.

Execução de migração de dados online usando o console

É possível usar o Console de gerenciamento da AWS para migrar os dados do cluster para o cluster do Valkey ou Redis OSS.

Como executar a migração de dados online usando o console

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Crie um novo cluster do Valkey ou Redis OSS ou escolha um cluster existente. Verifique se o cluster atende aos seguintes requisitos:
 - A versão do seu mecanismo deve ser o Valkey 7.2 ou posterior ou o Redis OSS 5.0.6 ou posterior.
 - O cluster não deve ter o AUTH habilitado.
 - O config `protected-mode` deve estar definido como `no`.
 - Se você tiver a configuração `bind` no config do Valkey ou Redis OSS, ela deve ser atualizada para permitir solicitações de nós do ElastiCache.
 - O número de bancos de dados deve ser o mesmo entre o nó do ElastiCache e o cluster do Valkey ou Redis OSS. Esse valor é definido usando `databases` no config do mecanismo.

- Os comandos do Valkey ou Redis OSS que executam a modificação de dados não devem ser renomeados para permitir que a replicação dos dados seja bem-sucedida.
 - Para replicar os dados do cluster do Valkey ou Redis OSS para o ElastiCache, verifique se há CPU e memória suficientes para lidar com essa carga adicional. Essa carga vem do arquivo RDB criado pelo cluster do Valkey ou Redis OSS e transferido pela rede para o nó do ElastiCache.
 - O cluster está no status available (disponível).
3. Com o cluster selecionado, escolha Migrate Data from Endpoint (Migrar dados do endpoint) para Actions (Ações).
 4. Na caixa de diálogo Migrar dados do endpoint, insira o endereço IP e a porta onde o cluster do Valkey ou Redis OSS está disponível.

 Important

O endereço IP deve ser exato. Se você inserir o endereço de maneira incorreta, a migração falhará.

5. Selecione Start Migration (Iniciar migração).

À medida que o cluster inicia a migração, ele muda para o status Modifying (Modificando) e, depois, para Migrating (Migrando).

6. Monitore o andamento da migração selecionando Events (Eventos) no painel de navegação.

A qualquer momento durante o processo de migração, é possível interromper a migração. Para fazer isso, escolha o cluster e selecione Stop Data Migration (Interromper a migração de dados) para Actions (Ações). Então, o cluster entrará no status Available (Disponível).

Se a migração for bem-sucedida, o cluster entrará no status Available (Disponível) e o log de eventos mostrará o seguinte:

Migration operation succeeded for replication group *ElastiCacheClusterName*.

Se a migração falhar, o cluster entrará no status Available (Disponível) e o log de eventos mostrará o seguinte:

Migration operation failed for replication group *ElastiCacheClusterName*.

Seleção de regiões e zonas de disponibilidade para o ElastiCache

Você pode oferecer escalabilidade e confiabilidade adicionais a clusters do ElastiCache designando regiões e zonas de disponibilidade usando o endpoint correspondente.

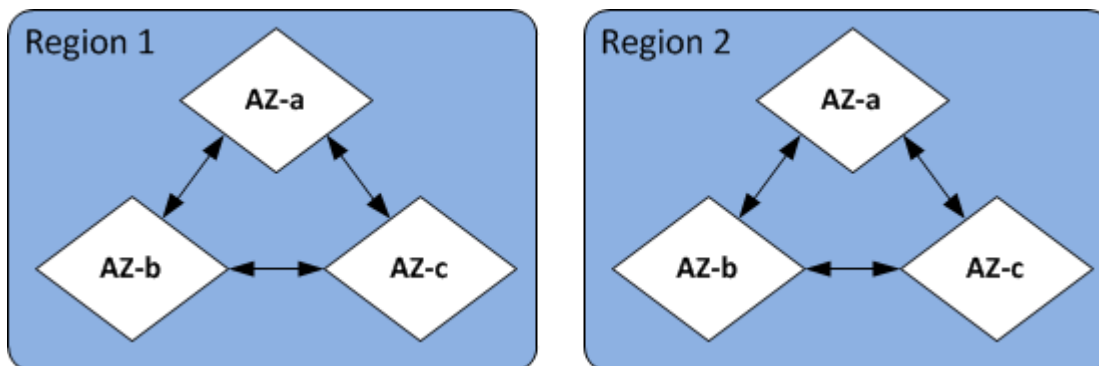
Os recursos de computação em nuvem da AWS são abrigados em instalações de data centers de alta disponibilidade. Para fornecer escalabilidade e confiabilidade adicionais, estas instalações do datacenter estão localizadas em diferentes locais físicos. Esses locais são categorizados por regiões e zonas de disponibilidade.

ASWAs regiões da são grandes e amplamente dispersas em locais geográficos separados. As zonas de disponibilidade são diferentes localizações dentro de uma região da AWS que são projetadas para serem isoladas de falhas em outras zonas de disponibilidade. Elas fornecem conectividade de rede de baixa latência e custo reduzido para outras zonas de disponibilidade na mesma região da AWS.

Important

Cada região é totalmente independente. Qualquer atividade do ElastiCache iniciada (por exemplo, criação de clusters) é executada somente na região padrão atual.

Para criar ou trabalhar com um cluster em uma região específica, use o endpoint do serviço regional correspondente. Para os endpoints de serviço, consulte [Regiões e endpoints com suporte](#).



Regiões e zonas de disponibilidade da

Tópicos

- [Considerações sobre zonas de disponibilidade com o Memcached](#)
- [Localização dos seus nós](#)

- [Regiões e endpoints com suporte](#)
- [Uso de zonas locais com o ElastiCache](#)
- [Usando Outposts com ElastiCache](#)

Considerações sobre zonas de disponibilidade com o Memcached

A distribuição de nós do Memcached em várias zonas de disponibilidade em uma região ajuda a protegê-los contra o impacto de uma falha catastrófica, como perda de energia em uma zona de disponibilidade.

Armazenamento em cache sem servidor

O armazenamento em cache sem servidor do ElastiCache cria um cache altamente disponível que abrange várias zonas de disponibilidade. Você pode especificar sub-redes de zonas de disponibilidade diferentes e da mesma VPC ao criar o cluster de tecnologia sem servidor, ou o ElastiCache vai escolher automaticamente sub-redes da VPC padrão.

Projeto do próprio ElastiCache para cluster do Memcached

Um cluster do Memcached pode ter até 300 nós. Ao criar ou adicionar nós ao cluster do Memcached, você pode especificar uma única zona de disponibilidade para todos os seus nós, permitir que o ElastiCache escolha uma única zona de disponibilidade para todos os seus nós, especificar as zonas de disponibilidade para cada nó ou permitir que o ElastiCache escolha uma zona de disponibilidade para cada nó. Novos nós podem ser criados em diferentes zonas de disponibilidade à medida que você os adiciona a um cluster do Memcached existente. Depois que um nó de cache é criado, sua zona de disponibilidade não pode ser modificada.

Se você quiser que um cluster em um único cluster da zona de disponibilidade tenha seus nós distribuídos em várias zonas de disponibilidade, o ElastiCache pode criar novos nós nas várias zonas de disponibilidade. Em seguida, você pode excluir alguns ou todos os nós de cache originais. Recomendamos essa abordagem.

Para migrar nós do Memcached de uma única zona de disponibilidade para várias zonas de disponibilidade

1. Modifique seu cluster criando novos nós de cache nas zonas de disponibilidade onde você deseja. Na sua solicitação, faça o seguinte:
 - Defina AZMode (CLI: `- -az-mode`) como `cross-az`.

- Defina NumCacheNodes (CLI: - -num-cache-nodes) como o número de nós de cache ativos no momento mais o número de novos nós de cache que você deseja criar.
- Defina NewAvailabilityZones (CLI: - -new-availability-zones) como uma lista das zonas nas quais você deseja que os novos nós de cache sejam criados. Para permitir que o ElastiCache determine a zona de disponibilidade para cada novo nó, não especifique uma lista.
- Defina ApplyImmediately (CLI: - -apply-immediately) como verdadeiro.

 Note

Se você não estiver usando a descoberta automática, atualize sua aplicação cliente com os novos endpoints do nó de cache.

Antes de passar para a próxima etapa, certifique-se de que os nós do Memcached estão totalmente criados e disponíveis.

2. Modifique seu cluster removendo os nós que você não deseja mais na zona de disponibilidade original. Na sua solicitação, faça o seguinte:

- Defina NumCacheNodes (CLI: - -num-cache-nodes) como o número de nós de cache ativos que você deseja após esta modificação ser aplicada.
- Defina CacheNodeIdsToRemove (CLI: - -nodes-to-remove) como uma lista dos nós de cache que você deseja remover do cluster.

O número de IDs de nó de cache listados deve ser igual ao número de nós ativos no momento menos o valor em NumCacheNodes.

- (Opcional) Defina ApplyImmediately (CLI: - -apply-immediately) como verdadeiro.

Se você não definir ApplyImmediately (CLI: - -apply-immediately) como verdadeiro, as exclusões de nó ocorrerão na sua próxima janela de manutenção.

Localização dos seus nós

O Amazon ElastiCache oferece suporte para localizar todos os nós de um cluster em uma ou várias zonas de disponibilidade (AZs). Além disso, se você optar por localizar os nós em várias AZs (recomendado), o ElastiCache permitirá escolher a AZ para cada nó ou permitir que o ElastiCache a escolha para você.

Ao localizar os nós em diferentes AZs, você elimina a chance de que uma falha, como uma queda de energia, em uma AZ faça com que todo o seu sistema falhe. O teste demonstrou que não há diferença de latência significativa entre localizar todos os nós em uma AZ ou espalhá-los em várias AZs.

Você pode especificar uma AZ para cada nó ao criar um cluster ou adicionar nós ao modificar um cluster existente. Ao especificar uma AZ para cada nó ao criar um cluster, a AZ deve estar disponível nesse grupo de sub-redes. Para obter mais informações, consulte:

- [Criação de um cluster do Memcached](#)
- [Criação de um cluster do Valkey ou Redis OSS](#)
- [Modificação de um cluster do ElastiCache](#)
- [Adicionar nós a um ElastiCache cluster](#)

Regiões e endpoints com suporte

O Amazon ElastiCache está disponível em várias regiões da AWS. Isso significa que você pode iniciar clusters do ElastiCache nos locais que atendam aos seus requisitos. Por exemplo, você pode ativá-los na região da AWS mais próxima de seus clientes ou ativá-los em determinada região da AWS para atender a determinados requisitos legais.

Cada região é projetada para ser completamente isolada das outras. Em cada região há várias zonas de disponibilidade (AZ). Os caches de tecnologia sem servidor do ElastiCache replicam dados automaticamente em várias zonas de disponibilidade (exceto us-west-1, em que os dados são replicados em duas zonas de disponibilidade) para alta disponibilidade. Ao projetar o próprio cluster do ElastiCache, você pode optar por iniciar os nós em AZs diferentes para obter tolerância a falhas. Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#) no início deste tópico.

Regiões em que o ElastiCache tem suporte

Nome da região/região	Endpoint	Protocolo	
Região Leste dos EUA (Ohio) us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS	
Região Leste dos EUA (Norte da Virgínia) us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS	
Região Oeste dos EUA (Norte da Califórnia) us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS	
Região Oeste dos EUA (Oregon) us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS	
Região Canadá (Central) ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS	
Região do Canadá (Oeste) ca-west-1	elasticache.ca-west-1.amazonaws.com	HTTPS	
Ásia-Pacífico (Jacarta) ap-southeast-3	elasticache.ap-southeast-3.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo	
Região Ásia-Pacífico (Mumbai) ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS	
Ásia-Pacífico (Haiderabade) ap-south-2	elasticache.ap-south-2.amazonaws.com	HTTPS	
Região Ásia-Pacífico (Tóquio) ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS	
Região Ásia-Pacífico (Seul) ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS	
Região Ásia-Pacífico (Osaka) ap-northeast-3	elasticache.ap-northeast-3.amazonaws.com	HTTPS	
Região Ásia-Pacífico (Singapura) ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS	
Ásia-Pacífico (Sydney) ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo	
Região Europa (Frankfurt) eu-central-1	elasticache.eu-central-1.amazonaws.com	HTTPS	
Região Europa (Zurique) eu-central-2	elasticache.eu-central-2.amazonaws.com	HTTPS	
Região Europa (Estocolmo) eu-north-1	elasticache.eu-north-1.amazonaws.com	HTTPS	
Região Oriente Médio (Bahrein) me-south-1	elasticache.me-south-1.amazonaws.com	HTTPS	
Região do Oriente Médio (Emirados Árabes Unidos) me-central-1	elasticache.me-central-1.amazonaws.com	HTTPS	
Região Europa (Irlanda) eu-west-1	elasticache.eu-west-1.amazonaws.com	HTTPS	
Região Europa (Londres) eu-west-2	elasticache.eu-west-2.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo	
Região Europa (Paris) eu-west-3	elasticache.eu-west-3.amazonaws.com	HTTPS	
Região Europa (Milão) eu-south-1	elasticache.eu-south-1.amazonaws.com	HTTPS	
Região Europa (Espanha) eu-south-2	elasticache.eu-south-2.amazonaws.com	HTTPS	
Região América do Sul (São Paulo) sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS	
Região China (Pequim) cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS	
Região China (Ningxia) cn-northwest-1	elasticache.cn-northwest-1.amazonaws.com.cn	HTTPS	
Região Ásia-Pacífico (Hong Kong) ap-east-1	elasticache.ap-east-1.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo	
Região África (Cidade do Cabo) af-south-1	elasticache.af-south-1.amazonaws.com	HTTPS	
Região de Israel (Tel Aviv) il-central-1	elasticache.il-central-1.amazonaws.com	HTTPS	
AWS GovCloud (Oeste dos EUA) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS	
AWS GovCloud (Leste dos EUA) us-gov-east-1	elasticache.us-gov-east-1.amazonaws.com	HTTPS	

Para obter informações sobre o uso da AWS GovCloud (EUA) com o ElastiCache, consulte [Serviços na região da AWS GovCloud \(EUA\): ElastiCache](#).

Algumas regiões oferecem suporte a um subconjunto de tipos de nós. Para obter uma tabela dos tipos de nós compatíveis por região da AWS, consulte [Tipos de nós compatíveis por região da AWS](#).

A maioria das regiões é compatível com o estabelecimento de uma conexão privada entre a VPC e os endpoints da API do ElastiCache criando um endpoint da VPC da interface por meio do AWS PrivateLink. Para obter mais informações, consulte [ElastiCache API e interface VPC endpoints \(AWS PrivateLink\)](#).

Para obter uma tabela de produtos e serviços da AWS por região, consulte [Produtos e serviços por região](#).

Uso de zonas locais com o ElastiCache

Uma zona local é uma extensão de uma região AWS que está geograficamente próxima de seus usuários. É possível estender qualquer nuvem privada virtual (VPC) de uma região principal da AWS para as zonas locais criando uma nova sub-rede e atribuindo-a a uma Local Zone. Quando você criar uma sub-rede em uma zona local, sua VPC também será estendida para essa zona local. A sub-rede na zona local funciona da mesma forma que outras sub-redes na VPC.

Local Zones (Zonas locais) permitem que você coloque recursos como um cluster do ElastiCache em vários locais próximos aos seus usuários.

Quando criar um cluster do ElastiCache, você poderá escolher uma sub-rede em uma zona local. As zonas locais têm suas próprias conexões com a Internet e suporte no Direct Connect. Assim, os recursos criados em uma zona local podem atender usuários locais com comunicações de latência muito baixa. Para obter mais informações, consulte [Zonas locais da AWS](#).

Uma zona local é representada por um código de região da AWS seguido por um identificador que indica o local, por exemplo, `us-west-2-lax-1a`.

Neste momento, zonas locais disponíveis são `us-west-2-lax-1a` e `us-west-2-lax-1b`.

As seguintes limitações se aplicam ao ElastiCache para zonas locais:

- Não há suporte para os armazenamentos de dados globais.
- Não há suporte para migração online.
- Há suporte de zonas locais para os tipos de nó a seguir no momento:
 - Geração atual:

Tipos de nó M: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

Tipos de nó R: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

Tipos de nó T: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Ativação de uma zona local

1. Habilite a zona local no console do Amazon EC2.

Para obter mais informações, consulte [Habilitação de zonas locais](#) no Guia do usuário do Amazon EC2.

2. Crie uma sub-rede na zona local.

Para obter mais informações, consulte [Criar uma sub-rede na VPC](#) no Guia do usuário da Amazon VPC.

3. Criar um grupo de sub-redes do ElastiCache na zona local.

Ao criar um grupo de sub-redes do ElastiCache, escolha o grupo de zonas de disponibilidade para a zona local.

Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

4. Criar um cluster do ElastiCache for Memcached que use a sub-rede do ElastiCache na zona local.

Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).

5. Criar um cluster do ElastiCache for Redis OSS que use a sub-rede do ElastiCache na zona local. Para obter mais informações, consulte um dos tópicos a seguir:

- [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#)
- [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)

Usando Outposts com ElastiCache

Você pode usar AWS Outposts com ElastiCache Outposts. ElastiCache Outposts é um serviço totalmente gerenciado que estende a AWS infraestrutura APIs, os serviços e as ferramentas até as instalações do cliente. Ao fornecer acesso local à infraestrutura AWS gerenciada, o AWS Outposts permite que os clientes criem e executem aplicativos no local usando as mesmas interfaces de programação das AWS regiões, enquanto usam recursos locais de computação e armazenamento para reduzir a latência e as necessidades locais de processamento de dados. Um posto avançado é um pool de capacidade de AWS computação e armazenamento implantado no local do cliente. AWS opera, monitora e gerencia essa capacidade como parte de uma AWS região. Você pode criar sub-redes em seu Outpost e especificá-las ao criar AWS recursos, como clusters. ElastiCache

Note

Nesta versão, as seguintes limitações se aplicam:

- ElastiCache for Outposts oferece suporte apenas às famílias de nós M5 e R5.
- Multi-AZ (não há suporte para replicação cruzada do Outpost).
- Não há suporte para migração ao vivo.
- Não há suporte para snapshots locais.
- Os logs do mecanismo e os logs lentos não podem ser habilitados.
- ElastiCache on Outposts não suporta ColP.
- ElastiCache for Outposts não é suportado nas seguintes regiões: cn-north-1, cn-northwest-1 e ap-northeast-3.

Usando Outposts com o console ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione Caches do Valkey, Caches do Redis OSS ou Caches do Memcached.
3. Se você escolher Caches do Valkey, selecione Criar cache do Valkey. Se você escolher Caches do Redis OSS, selecione Criar cache do Redis OSS. Se você escolher Caches do Memcached, selecione Criar cache do Memcached.
4. Em Configurações de cluster, selecione Projetar seu próprio cache e Cache de cluster. Deixe o Modo Cluster definido como Desabilitado. Em seguida, crie um nome e uma descrição opcional para o cache.
5. Para localização, escolha On-premises.
6. Na seção On-premises, você verá o campo ID do Outpost. Insira o ID de onde o cluster será executado.

Todas as configurações adicionais em Configurações de cluster podem permanecer como padrão.

7. Em Conectividade, selecione Criar um novo grupo de sub-redes e insira o ID da VPC. Deixe o resto como padrão e escolha Próximo.

Configurar opções on-premises

Você pode selecionar um Outpost disponível para adicionar seu cluster ou, se não houver Outposts disponíveis, criar um novo usando as seguintes etapas:

Em On-Premises options (Opções on-premises):

1. Em Configurações do Valkey, Configurações do Redis OSS ou Configurações do Memcached, dependendo do seu mecanismo preferido:
 - a. Nome: insira um nome para o cluster
 - b. Descrição: insira uma descrição para o cluster.
 - c. Compatibilidade da versão do motor: a versão do motor é baseada na região Outpost AWS
 - d. Porta: para Valkey ou Redis OSS, aceite a porta padrão 6379. Para Memcached, aceite a porta padrão 11211. Se preferir usar uma porta diferente, digite o número da porta.
 - e. Parameter group (Grupo de parâmetros): use a lista suspensa para selecionar um grupo de parâmetros padrão ou personalizado.
 - f. Node Type (Tipo de nó): as instâncias disponíveis são baseadas na disponibilidade dos Outposts. Se estiver usando o Valkey ou o Redis OSS, o Assistente de Portabilidade para .NET para Outposts aceita apenas famílias de nós M5 e R5. Na lista suspensa, selecione Outposts e selecione um tipo de nó disponível que você deseje usar para esse cluster. Em seguida, selecione Save (Salvar).
 - g. Number of Replicas (Número de réplicas): insira o número de réplicas de leitura desejadas para este grupo de replicação. Você deve ter pelo menos uma e não mais de cinco réplicas de leitura. O valor padrão é 2.

Os nomes gerados automaticamente das réplicas de leitura seguem o mesmo padrão que o nome do cluster primário, com um traço e um número sequencial de três dígitos adicionados ao final, começando com -002. Por exemplo, se o grupo de replicação for denominado MyGroup, então os nomes dos secundários seriam MyGroup-002, MyGroup-003, MyGroup-004, MyGroup-005, MyGroup-006.

2. Em Connectivity (Conectividade):

- a. Subnet Group (Grupo de sub-redes): na lista, selecione Create new (Criar novo).
 - Name (Nome): insira um nome para o grupo de sub-redes.
 - Description (Descrição): insira uma descrição para o grupo de sub-redes.

- VPC ID (ID da VPC): o ID da VPC deve corresponder à VPC do Outpost. Se você selecionar uma VPC que não tenha sub-rede nos Outposts, IDs a lista retornará vazia.
- Availability Zone or Outpost (Zona de disponibilidade ou Outpost): selecione o Outpost que você está usando.
- Subnet ID (ID da sub-rede): selecione um ID de sub-rede disponível para o Outpost. Se não houver nenhuma sub-rede IDs disponível, você precisará criá-la. Para obter mais informações, consulte [Criar uma sub-rede](#).

b. Escolha Criar.

Visualização de detalhes do cluster do Outpost

Na página da lista, selecione um cluster que pertença a um AWS Posto Avançado e observe o seguinte ao visualizar os detalhes do Cluster:

- Zona de disponibilidade: representará o Posto Avançado, usando um ARN (Amazon Resource Name) e AWS o número do recurso.
- Nome do Posto Avançado: O nome do Posto AWS Avançado.

Usando Outposts com a CLI AWS

Você pode usar o AWS Command Line Interface(AWS CLI) para controlar vários AWS serviços na linha de comando e automatizá-los por meio de scripts. Você pode usar a AWS CLI para operações ad hoc (únicas).

Baixando e configurando o AWS CLI

AWS CLIÉ executado em Windows, macOS ou Linux. Use o procedimento a seguir para fazer download e configurá-la.

Como fazer download, instalar e configurar a CLI

1. Faça o download da AWS CLI na página da Web da [interface de linha de AWS comando](#).
2. Siga as instruções para [instalar a AWS CLI](#) e [configurar a AWS CLI no Guia do](#) usuário.AWS Command Line Interface

Usando a AWS CLI com Outposts

Use a seguinte operação da CLI para criar um cluster que use Outposts:

- [create-cache-cluster](#)— Usando essa operação, o `outpost-mode` parâmetro aceita um valor que especifica se os nós no cluster são criados em um único Outpost ou em vários Outposts.

Note

No momento, somente há suporte para o modo `single-outpost`.

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id cluster id \  
  --outpost-mode single-outpost \  
  
```

Trabalhando com ElastiCache

Nesta seção, você pode encontrar detalhes sobre como gerenciar os vários componentes da sua ElastiCache implementação.

Tópicos

- [Snapshots e restauração](#)
- [Versões e atualizações do motor em ElastiCache](#)
- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)
- [Gerenciando seu cluster baseado em nós no ElastiCache](#)
- [Conectar uma instância do EC2 e um cache do ElastiCache automaticamente](#)
- [Dimensionamento ElastiCache](#)
- [Conceitos básicos do filtro Bloom](#)
- [Introdução ao Watch em Serverless](#)
- [Conceitos básicos de pesquisa vetorial](#)
- [Conceitos básicos do JSON para Valkey e Redis OSS](#)
- [Marcando seus recursos ElastiCache](#)
- [Usar a Lente do Well-Architected para o Amazon ElastiCache](#)
- [Etapas comuns de solução de problemas e melhores práticas com ElastiCache](#)

Snapshots e restauração

ElastiCache Os caches da Amazon que executam Valkey, Redis OSS ou Serverless Memcached podem fazer backup de seus dados criando um snapshot. É possível usar o backup para restaurar um cache ou propagar dados para um novo cache. O backup consiste nos metadados do cache com todos os dados no cache. Todos os backups são gravados no Amazon Simple Storage Service (Amazon S3), que fornece armazenamento durável. A qualquer momento, você pode restaurar os dados criando um novo cache do Valkey, Redis OSS ou Memcached de tecnologia sem servidor e o preenchendo com dados de um backup. Com ElastiCache, você pode gerenciar backups usando o Console de gerenciamento da AWS, o AWS Command Line Interface(AWS CLI) e a ElastiCache API.

Se quiser excluir um cache e for importante preservar os dados, você poderá tomar uma precaução extra. Para isso, crie um backup manual primeiro, verifique se o status está disponível e, em seguida, exclua o cluster. Isso garante que, se o backup falhar, você ainda terá os dados do cache

disponíveis. Você pode tentar um backup novamente, seguindo as práticas recomendadas descritas anteriormente.

Tópicos

- [Restrições do backup](#)
- [Impacto sobre o desempenho dos backups de clusters baseados em nós](#)
- [Programação de backups automáticos](#)
- [Realização de backups manuais](#)
- [Criação de um backup final](#)
- [Descrição de backups](#)
- [Copiar backups](#)
- [Exportação de um backup](#)
- [Restauração de um backup para um novo cache](#)
- [Exclusão de um backup](#)
- [Marcação de backups](#)
- [Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente](#)

Restrições do backup

Considere as seguintes restrições ao planejar ou fazer backups:

- O backup e a restauração são suportados apenas para caches executados no Valkey, Redis OSS ou Memcached de tecnologia sem servidor.
- Para clusters do Valkey ou Redis OSS (modo cluster desabilitado), não há suporte para backup e restauração em nós `cache.t1.micro`. Todos os outros tipos de nó de cache têm suporte.
- Para clusters do Valkey ou Redis OSS (modo cluster habilitado), o backup e a restauração têm suporte para todos os tipos de nó.
- Durante qualquer período contíguo de 24 horas, você não pode criar mais de 24 backups manuais por cache de tecnologia sem servidor. Para clusters baseados em nós do Valkey e Redis OSS, você não pode criar mais de 20 backups manuais por nó do cluster.
- O Valkey ou Redis OSS (modo cluster habilitado) oferece suporte apenas para backups no nível do cluster (para a API ou CLI, no nível do grupo de replicação). O Valkey ou Redis OSS (modo cluster habilitado) não oferece suporte à execução de backups no nível do fragmento (para a API ou CLI, no nível do grupo de nós).

- Durante o processo de backup, não é possível executar outra operação de API ou CLI no cache de tecnologia sem servidor. Você pode executar operações de API ou CLI em um cluster baseado em nós durante o backup.
- Se você estiver usando caches do Valkey ou Redis OSS com classificação de dados em níveis, não será possível exportar um backup para o Amazon S3.
- Só é possível restaurar um backup de um cluster usando o tipo de nó r6gd para clusters que estejam usando o tipo de nó r6gd.

Impacto sobre o desempenho dos backups de clusters baseados em nós

Os backups em caches sem servidor são transparentes para a aplicação, sem impacto sobre o desempenho. No entanto, durante a criação de backups para clusters baseados em nós, é possível que haja algum impacto sobre o desempenho, dependendo da memória reservada disponível. Os backups para clusters baseados em nós não estão disponíveis com ElastiCache o Memcached, mas estão disponíveis com ElastiCache o Redis OSS.

Estas são as diretrizes a fim de melhorar o desempenho do backup para clusters baseados em nós.

- Definir o `reserved-memory-percent` parâmetro — Para reduzir a paginação excessiva, recomendamos que você defina o `reserved-memory-percent` parâmetro. Esse parâmetro impede que o Valkey e Redis OSS consumam toda a memória disponível do nó e pode ajudar a reduzir a quantidade de paginação. Você também pode ver melhorias de desempenho simplesmente usando um nó maior. Para obter mais informações sobre a memória reservada e `reserved-memory-percent` parâmetros, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).
- Crie backups a partir de uma réplica de leitura: se você estiver executando o Valkey ou Redis OSS em um grupo de nós com mais de um nó, será possível fazer um backup do nó primário ou de uma das réplicas de leitura. Devido aos recursos do sistema necessários durante um BGSAVE, recomendamos que você crie backups de uma das réplicas de leitura. Enquanto o backup está sendo criado da réplica, o nó primário permanece inalterado pelos requisitos de recursos de BGSAVE. O nó primário pode continuar a atender solicitações sem ficar lento.

Para fazer isso, consulte [Criação de um backup manual \(console\)](#), e no campo Cluster Name (Nome do cluster) na janela Create Backup (Criar backup), escolha uma réplica em vez do nó primário padrão.

Se você excluir um grupo de replicação e solicitar um backup final, ElastiCache sempre fará o backup do nó primário. Isso garante que você capture os dados mais recentes do Valkey ou Redis OSS antes que o grupo de replicação seja excluído.

Programação de backups automáticos

É possível habilitar backups automáticos para qualquer cache de tecnologia sem servidor do Valkey ou Redis OSS ou para qualquer cluster baseado em nós. Quando os backups automáticos estão habilitados, ElastiCache cria um backup do cache diariamente. Não há impacto sobre o cache, e a alteração é imediata. Os backups automáticos podem ajudar a proteger contra a perda de dados. Em caso de falha, você pode criar um cache e restaurar os dados usando o backup mais recente. O resultado é um cache inicializado a quente, pré-carregado com os dados e pronto para uso. Para obter mais informações, consulte [Restauração de um backup para um novo cache](#).

É possível ativar backups automáticos para qualquer cache de tecnologia sem servidor do Memcached. Quando os backups automáticos estão habilitados, ElastiCache cria um backup do cache diariamente. Não há impacto sobre o cache, e a alteração é imediata. Os backups automáticos podem ajudar a proteger contra a perda de dados. Em caso de falha, você pode criar um cache e restaurar os dados usando o backup mais recente. O resultado é um cache inicializado a quente, pré-carregado com os dados e pronto para uso. Para obter mais informações, consulte [Restauração de um backup para um novo cache](#).

Ao agendar backups automáticos, você deve planejar as seguintes configurações:

- Hora de início do backup — Uma hora do dia em que ElastiCache começa a criar um backup. É possível definir a janela de backup para qualquer momento quando mais prático. Se você não especificar uma janela de backup, ElastiCache atribuirá uma automaticamente.
- Backup retention limit (Limite de retenção de backup): o número de dias em que o backup é mantido no Amazon S3. Por exemplo, se o limite de retenção for definido como 5, um backup feito hoje será mantido por 5 dias. Quando o limite de retenção expirar, o backup será excluído automaticamente.

O limite máximo de retenção de backups é de 35 dias. Se o limite de retenção do backup for definido como 0, os backups automáticos serão desabilitados para o cache.

Quando você agendar backups automáticos, ElastiCache começará a criar o backup. É possível definir a janela de backup para qualquer momento quando mais prático. Se você não especificar uma janela de backup, ElastiCache atribuirá uma automaticamente.

Você pode ativar ou desativar os backups automáticos ao criar um novo cache ou atualizar um cache existente usando o ElastiCache console AWS CLI, o ou a ElastiCache API. Para Valkey e Redis OSS, isso é feito marcando a caixa Ativar backups automáticos na seção Configurações avançadas do Valkey ou Configurações avançadas do Redis OSS. Para Memcached, isso é feito marcando a caixa Ativar backups automáticos na seção Configurações avançadas do Memcached.

Realização de backups manuais

Além dos backups automáticos, você pode criar um backup manual a qualquer momento. Ao contrário dos backups automáticos, que são excluídos automaticamente após um período de retenção especificado, os backups manuais não têm um período de retenção após o qual são excluídos automaticamente. Mesmo se você excluir o cache, todos os backups manuais desse cache são retidos. Caso não queira mais manter um backup manual, você deverá excluí-lo explicitamente por conta própria.

Além de criar diretamente um backup manual, você pode criar um backup manual de uma das seguintes maneiras:

- [Copiar backups](#). Não importa se o backup de origem foi criado automática ou manualmente.
- [Criação de um backup final](#). Crie imediatamente um backup antes de excluir um cluster ou nó.

Você pode criar um backup manual de um cache usando a Console de gerenciamento da AWS, a CLI, a ou a ElastiCache API.

Você pode gerar backups manuais a partir de réplicas com o modo de cluster habilitado e o modo de cluster desabilitado.

Criação de um backup manual (console)

Para criar um backup de um cache (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Caches do Valkey, Caches do Redis OSS ou Caches do Memcached, dependendo da sua preferência.
3. Selecione a caixa à esquerda do nome do cache do qual você deseja fazer backup.
4. Escolha Backup.
5. No diálogo Criar backup, digite um nome para seu backup na caixa Nome do backup. Recomendamos que o nome indique o cluster do backup e a data e hora de criação do backup.

As restrições de nomenclatura de cluster são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.

- Não podem conter dois hífens consecutivos.
- Não podem terminar com um hífen.

6. Escolha Create Backup.

O status do cluster muda para snapshotting.

Criação de um backup manual (AWS CLI)

Backup manual de um cache sem servidor com o AWS CLI

Para criar um backup manual de um cache usando o AWS CLI, use a `create-serverless-snapshot` AWS CLI operação com os seguintes parâmetros:

- `--serverless-cache-name`: o nome do cache sem servidor do qual você está fazendo backup.
- `--serverless-cache-snapshot-name` - Nome do snapshot a ser criado.

Para Linux, macOS ou Unix:

```
aws elasticache create-serverless-snapshot \  
    --serverless-cache-name CacheName \  
    --serverless-cache-snapshot-name bkup-20231127
```

Para Windows:

```
aws elasticache create-serverless-snapshot ^  
    --serverless-cache-name CacheName ^  
    --serverless-cache-snapshot-name bkup-20231127
```

Backup manual de um cluster baseado em nós com o AWS CLI

Para criar um backup manual de um cluster baseado em nós usando o AWS CLI, use a `create-snapshot` AWS CLI operação com os seguintes parâmetros:

- `--cache-cluster-id`
 - Se o cluster do qual você está fazendo backup não tiver nós de réplica, `--cache-cluster-id` é o nome do cluster do qual você está fazendo backup, por exemplo. *mycluster*

- Se o cluster do backup tiver um ou mais nós de réplica, `--cache-cluster-id` será o nome do nó no cluster que você deseja usar para o backup. Por exemplo, o nome pode ser `mycluster-002`.

Use este parâmetro somente ao fazer backup de um cluster do Valkey ou Redis OSS (modo cluster desabilitado).

- `--replication-group-id`: nome do cluster do Valkey ou Redis OSS (modo cluster habilitado) (CLI/API: um grupo de replicação) para usar como fonte para o backup. Use este parâmetro ao fazer backup de um cluster do Valkey ou Redis OSS (modo cluster habilitado).
- `--snapshot-name` - Nome do snapshot a ser criado.

As restrições de nomenclatura de cluster são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

Exemplo 1: backup de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) que não tem nós de réplicas

A AWS CLI operação a seguir cria o backup `bkup-20150515` do cluster Valkey ou Redis OSS (modo de cluster desativado) `myNonClusteredRedis` que não tem réplicas de leitura.

Para Linux, macOS ou Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis \  
  --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis ^  
  --snapshot-name bkup-20150515
```

Exemplo 2: backup de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com nós de réplicas

A AWS CLI operação a seguir cria o backup `bkup-20150515` do cluster Valkey ou Redis OSS (modo de cluster desativado). `myNonClusteredRedis` Esse backup tem uma ou mais réplicas de leitura.

Para Linux, macOS ou Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis-001 \  
  --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis-001 ^  
  --snapshot-name bkup-20150515
```

Saída do exemplo: backup de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com nós de réplicas

A saída da operação é semelhante à seguinte.

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x",  
    "VpcId": "vpc-91280df6",  
    "CacheClusterId": "myNonClusteredRedis-001",  
    "SnapshotRetentionLimit": 0,  
    "NumCacheNodes": 1,  
    "SnapshotName": "bkup-20150515",  
    "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",  
    "AutoMinorVersionUpgrade": true,  
    "PreferredAvailabilityZone": "us-east-1c",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "SnapshotWindow": "08:30-09:30",  
    "EngineVersion": "6.0",  
    "NodeSnapshots": [  
      {
```

```

        "CacheSize": "",
        "CacheNodeId": "0001",
        "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"
    }
],
"CacheSubnetGroupName": "default",
"Port": 6379,
"PreferredMaintenanceWindow": "wed:07:30-wed:08:30",
"CacheNodeType": "cache.m3.2xlarge",
"DataTiering": "disabled"
}
}

```

Exemplo 3: backup de um cluster do Valkey ou Redis OSS (modo cluster habilitado)

A AWS CLI operação a seguir cria o backup `bkup-20150515` do cluster Valkey ou Redis OSS (modo de cluster ativado). `myClusteredRedis` Observe o uso de `--replication-group-id` no lugar de `--cache-cluster-id` para identificar a origem. Observe também que ElastiCache faz o backup usando o nó de réplica quando presente e usará como padrão o nó primário se um nó de réplica não estiver disponível.

Para Linux, macOS ou Unix:

```

aws elasticache create-snapshot \
  --replication-group-id myClusteredRedis \
  --snapshot-name bkup-20150515

```

Para Windows:

```

aws elasticache create-snapshot ^
  --replication-group-id myClusteredRedis ^
  --snapshot-name bkup-20150515

```

Saída do exemplo: backup de um cluster do Valkey ou Redis OSS (modo cluster habilitado)

A saída dessa operação é semelhante à seguinte.

```

{
  "Snapshot": {
    "Engine": "redis",
    "CacheParameterGroupName": "default.redis6.x.cluster.on",

```

```
"VpcId": "vpc-91280df6",
"NodeSnapshots": [
  {
    "CacheSize": "",
    "NodeGroupId": "0001"
  },
  {
    "CacheSize": "",
    "NodeGroupId": "0002"
  }
],
"NumNodeGroups": 2,
"SnapshotName": "bkup-20150515",
"ReplicationGroupId": "myClusteredRedis",
"AutoMinorVersionUpgrade": true,
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotStatus": "creating",
"SnapshotSource": "manual",
"SnapshotWindow": "10:00-11:00",
"EngineVersion": "6.0",
"CacheSubnetGroupName": "default",
"ReplicationGroupDescription": "2 shards 2 nodes each",
"Port": 6379,
"PreferredMaintenanceWindow": "sat:03:30-sat:04:30",
"CacheNodeType": "cache.r3.large",
"DataTiering": "disabled"
}
```

Tópicos relacionados

Para obter mais informações, consulte [create-snapshot](#) na Referência de comandos da AWS CLI.

Criando um backup usando CloudFormation

Você pode usar CloudFormation para criar um backup do seu cache ElastiCache Redis OSS ou Valkey, usando as `AWS::ElastiCache::ServerlessCache` propriedades ou `AWS::ElastiCache::ReplicationGroup`.

Usar o recurso **AWS::ElastiCache::ServerlessCache**

Use isso para criar um backup usando o `AWS::ElastiCache::ServerlessCache` recurso:

Resources:

```
    iotCatalog:
      Type: AWS::ElastiCache::ServerlessCache
      Properties:
        ...
        ServerlessCacheName: "your-cache-name"
        Engine: "redis"
        CacheUsageLimits
```

Usando o `AWS::ElastiCache::ReplicationGroup` recurso

Use o recurso `AWS::ElastiCache::ReplicationGroup`:

Resources:

```
    iotCatalog:
      Type: AWS::ElastiCache::ReplicationGroup
      Properties:
        ...
        ReplicationGroupDescription: "Description of your
replication group"

      Engine: "redis"
      CacheNodeType
      NumCacheClusters
      AutomaticFailoverEnabled
      AtRestEncryptionEnabled
```

Criação de um backup final

Você pode criar um backup final usando o ElastiCache console AWS CLI, o ou a ElastiCache API.

Criação de um backup final (console)

Você pode criar um backup final ao excluir um cache sem servidor Valkey, Memcached ou Redis OSS, ou um cluster baseado em nós Valkey ou Redis OSS, usando o console. ElastiCache

Para criar um backup final ao excluir um cache, na caixa de diálogo, escolha Sim, em Criar backup e dê um nome ao backup.

Tópicos relacionados

- [Usando o Console de gerenciamento da AWS](#)
- [Exclusão de um grupo de replicação \(Console\)](#)

Criação de um backup final (AWS CLI)

Você pode criar um backup final ao excluir um cache usando a AWS CLI.

Tópicos

- [Ao excluir um cache do Valkey, cache de tecnologia sem servidor do Memcached ou cache do Redis OSS](#)
- [Ao excluir um cluster do Valkey ou Redis OSS sem réplicas de leitura](#)
- [Ao excluir um cluster do Valkey ou Redis OSS com réplicas de leitura](#)

Ao excluir um cache do Valkey, cache de tecnologia sem servidor do Memcached ou cache do Redis OSS

Para criar um backup final, use a `delete-serverless-cache` AWS CLI operação com os seguintes parâmetros.

- `--serverless-cache-name`: nome do cache que está sendo excluído.
- `--final-snapshot-name` – Nome do backup.

O código a seguir cria o backup final `bkup-20231127-final` ao excluir o cache `myserverlesscache`.

Para Linux, macOS ou Unix:

```
aws elasticache delete-serverless-cache \  
    --serverless-cache-name myserverlesscache \  
    --final-snapshot-name bkup-20231127-final
```

Para Windows:

```
aws elasticache delete-serverless-cache ^  
    --serverless-cache-name myserverlesscache ^  
    --final-snapshot-name bkup-20231127-final
```

Para obter mais informações, consulte [delete-serverless-cache](#) na Referência de comandos da AWS CLI.

Ao excluir um cluster do Valkey ou Redis OSS sem réplicas de leitura

Para criar um backup final para um cluster baseado em nós sem réplicas de leitura, use a `delete-cache-cluster` AWS CLI operação com os parâmetros a seguir.

- `--cache-cluster-id`: nome do cluster que está sendo excluído.
- `--final-snapshot-identifier` – Nome do backup.

O código a seguir cria o backup final `bkup-20150515-final` ao excluir o cluster `myRedisCluster`.

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-cluster \  
    --cache-cluster-id myRedisCluster \  
    --final-snapshot-identifier bkup-20150515-final
```

Para Windows:

```
aws elasticache delete-cache-cluster ^  
    --cache-cluster-id myRedisCluster ^  
    --final-snapshot-identifier bkup-20150515-final
```

Para obter mais informações, consulte [delete-cache-cluster](#) na Referência de comandos da AWS CLI.

Ao excluir um cluster do Valkey ou Redis OSS com réplicas de leitura

Para criar um backup final ao excluir um grupo de replicação, use a `delete-replication-group` AWS CLI operação, com os seguintes parâmetros:

- `--replication-group-id` – Nome do grupo de replicação que está sendo excluído.
- `--final-snapshot-identifier` – Nome do backup final.

O código a seguir usa o backup final `bkup-20150515-final` ao excluir o grupo de replicação `myReplGroup`.

Para Linux, macOS ou Unix:

```
aws elasticache delete-replication-group \  
    --replication-group-id myReplGroup \  
    --final-snapshot-identifier bkup-20150515-final
```

Para Windows:

```
aws elasticache delete-replication-group ^  
    --replication-group-id myReplGroup ^  
    --final-snapshot-identifier bkup-20150515-final
```

Para obter mais informações, consulte [delete-replication-group](#) na Referência de comandos da AWS CLI.

Descrição de backups

Os procedimentos a seguir mostram como exibir uma lista dos seus backups. Se desejar, você também pode visualizar os detalhes de um backup específico.

Descrição de backups (console)

Para exibir backups usando o Console de gerenciamento da AWS

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Para ver os detalhes de um backup específico, escolha a caixa à esquerda do nome do backup.

Descrição dos backups sem servidor (AWS CLI)

Para exibir uma lista de backups sem servidor e, como opção, detalhes sobre um backup específico, use a operação `describe-serverless-cache-snapshots` da CLI.

Exemplos

A seguinte operação usa o parâmetro `--max-records` para listar até 20 backups associados à sua conta. Omitir o parâmetro `--max-records` lista até 50 backups.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

A operação a seguir usa o parâmetro `--serverless-cache-name` para só listar os backups associados ao cache `my-cache`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

A operação a seguir usa o parâmetro `--serverless-cache-snapshot-name` para exibir os detalhes do backup `my-backup`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

Para obter mais informações, consulte [describe-serverless-cache-snapshots](#) na Referência de AWS CLI Comandos.

Descrever backups de cluster baseados em nós (AWS CLI)

Para exibir uma lista de backups de cluster baseados em nós e, opcionalmente, detalhes sobre um backup específico, use a operação `describe-snapshots` da CLI.

Exemplos

A seguinte operação usa o parâmetro `--max-records` para listar até 20 backups associados à sua conta. Omitir o parâmetro `--max-records` lista até 50 backups.

```
aws elasticache describe-snapshots --max-records 20
```

A operação a seguir usa o parâmetro `--cache-cluster-id` para listar apenas os backups associados ao cluster `my-cluster`.

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

A operação a seguir usa o parâmetro `--snapshot-name` para exibir os detalhes do backup `my-backup`.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Para obter mais informações, consulte [describe-snapshots](#) na Referência de Comandos.AWS CLI

Copiar backups

É possível fazer uma cópia de qualquer backup, seja ele criado automática ou manualmente. Você também pode exportar seu backup para acessá-lo de fora ElastiCache. Para obter orientação sobre como exportar o backup, consulte [Exportação de um backup](#).

As etapas a seguir mostram como copiar um backup.

Cópia de backups (console)

Para copiar um backup (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista dos seus backups, no painel de navegação esquerdo, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup que você deseja copiar.
4. Escolha Ações e Copiar.
5. Na caixa New backup name, digite um nome para seu novo backup.
6. Escolha Copiar.

Cópia de um backup sem servidor (AWS CLI)

Para copiar um backup de um cache sem servidor, use a operação `copy-serverless-cache-snapshot`.

Parâmetros

- `--source-serverless-cache-snapshot-name` - Nome do backup a ser copiado.
- `--target-serverless-cache-snapshot-name` - Nome da cópia do backup.

O exemplo a seguir faz uma cópia de um backup automático.

Para 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
```

Para 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
```

Para ter mais informações, consulte [copy-serverless-cache-snapshot](#) no AWS CLI.

Copiar um backup de cluster baseado em nós (AWS CLI)

Para copiar um backup de um cluster baseado em nós, use a operação copy-snapshot.

Parâmetros

- `--source-snapshot-name` - Nome do backup a ser copiado.
- `--target-snapshot-name` - Nome da cópia do backup.
- `--target-bucket`: reservado para exportação de um backup. Não use esse parâmetro ao fazer uma cópia de um backup. Para obter mais informações, consulte [Exportação de um backup](#).

O exemplo a seguir faz uma cópia de um backup automático.

Para Linux, macOS ou Unix:

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \  
  --target-snapshot-name amzn-s3-demo-bucket
```

Para Windows:

```
aws elasticache copy-snapshot ^  
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^  
  --target-snapshot-name amzn-s3-demo-bucket
```

Para ter mais informações, consulte [copy-snapshot](#) no AWS CLI.

Exportação de um backup

A Amazon ElastiCache suporta a exportação de seu backup do OSS ElastiCache para Redis para um bucket do Amazon Simple Storage Service (Amazon S3), que dá acesso externo a ele. ElastiCache Você pode exportar um backup usando o ElastiCache console AWS CLI, o ou a ElastiCache API.

Exportar um backup pode ser útil se você precisar iniciar um cluster em outra AWS região. Você pode exportar seus dados em uma AWS região, copiar o arquivo.rdb para a nova AWS região e, em seguida, usar esse arquivo.rdb para alimentar o novo cache em vez de esperar que o novo cluster seja preenchido por meio do uso. Para obter informações sobre como criar um novo cluster, consulte [Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente](#). Outra razão pela qual você pode querer exportar os dados do seu cache é para usar o arquivo .rdb para processamento off-line.

Important

- O ElastiCache backup e o bucket do Amazon S3 para o qual você deseja copiá-lo devem estar na mesma AWS região.

Embora os backups copiados para um bucket do Amazon S3 sejam criptografados, recomendamos que você não conceda a outras pessoas acesso ao bucket do Amazon S3 no qual deseja armazená-los.

- A exportação de um backup para o Amazon S3 não é compatível com clusters que usam classificação de dados em níveis. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).
- A exportação de um backup está disponível para: clusters Valkey baseados em nós, clusters Redis OSS baseados em nós e caches com tecnologia sem servidor do Valkey, Memcached e Redis OSS. A exportação de um backup não está disponível para clusters baseados em nós do Memcached.

Antes de exportar um backup para um bucket do Amazon S3, você deve ter um bucket do Amazon S3 na AWS mesma região do backup. Conceda ElastiCache acesso ao bucket. As duas primeiras etapas mostram como fazer isso.

Crie um bucket Amazon S3

As etapas a seguir usam o console do Amazon S3 para criar um bucket do Amazon S3 onde você exporta e armazena seu backup. ElastiCache

Como criar um bucket do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Escolha Criar bucket.
3. Em Create a Bucket - Select a Bucket Name and Region, faça o seguinte:
 - a. Em Bucket Name (Nome do bucket), digite um nome para o bucket do Amazon S3.

O nome do bucket do Amazon S3 deve estar em conformidade com o DNS. Caso contrário, não ElastiCache conseguirá acessar seu arquivo de backup. As regras para a conformidade de DNS são:

- Os nomes devem ter no mínimo 3 e no máximo 63 caracteres de extensão.
 - Os nomes devem ser uma série de um ou mais rótulos separados por um ponto (.) em que cada rótulo:
 - Começa com uma letra minúscula ou um número.
 - Termina com uma letra minúscula ou um número.
 - Contém somente letras minúsculas, números e traços.
 - Os nomes não podem ser formatado como um endereço IP (por exemplo, 192.0.2.0).
- b. Na lista de regiões, escolha uma AWS região para seu bucket do Amazon S3. Essa AWS região deve ser a mesma AWS região do ElastiCache backup que você deseja exportar.
 - c. Escolha Criar.

Para obter mais informações sobre como criar um bucket do Amazon S3, consulte [Criação de um bucket](#), no Guia do usuário do Amazon Simple Storage Service.

Conceda ElastiCache acesso ao seu bucket Amazon S3

ElastiCache Para poder copiar um snapshot em um bucket do Amazon S3, você deve atualizar sua política de bucket do IAM para ElastiCache conceder acesso ao bucket.

 Warning

Mesmo que os backups copiados em um bucket do Amazon S3 sejam criptografados, seus dados podem ser acessados por qualquer pessoa com acesso ao seu bucket do Amazon S3. Portanto, recomendamos que você configure políticas do IAM para impedir o acesso não autorizado a esse bucket do Amazon S3. Para obter mais informações, consulte [Gerenciamento de acesso](#) no Guia do usuário do Amazon S3.

Para criar as permissões adequadas em um bucket do Amazon S3, siga as etapas descritas abaixo.

Para conceder ElastiCache acesso a um bucket do S3

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Escolha o nome do bucket do Amazon S3 para o qual você deseja copiar o backup. Esse deve ser o bucket do S3 que você criou em [Crie um bucket Amazon S3](#).
3. Selecione a guia Permissions (Permissões) e, em Permissions (Permissões), escolha Access control list (ACL) (Lista de controle de acesso (ACL)) e, depois, Edit (Editar).
4. Adicionar ID canônico do beneficiário
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353 com as seguintes opções:
 - Objetos: listar, gravar
 - ACL de bucket: ler, gravar

 Note

- Para a GovCloud região PDT, o ID canônico é.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Para a GovCloud região OSU, o ID canônico é.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Escolha Salvar.

Exportar e ElastiCache fazer backup

Agora você criou seu bucket do S3 e concedeu ElastiCache permissões para acessá-lo. Em seguida, você pode usar o ElastiCache console, a AWS CLI ou a ElastiCache API para exportar seu snapshot para ele.

Veja a seguir um exemplo de como a política atualizada pode parecer.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399484",
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    }
  ]
}
```

Para escolher regiões, veja a seguir um exemplo da aparência da política atualizada do IAM para o bucket do S3. O exemplo usa a região Ásia-Pacífico (Hong Kong).

JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
```

```
        "Action": "s3:*",
        "Resource": [
            "arn:aws:s3:::hkg-elasticache-backup",
            "arn:aws:s3:::hkg-elasticache-backup/*"
        ]
    },
    {
        "Sid": "Stmt15399484",
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": [
            "arn:aws:s3:::hkg-elasticache-backup",
            "arn:aws:s3:::hkg-elasticache-backup/*"
        ]
    }
]
```

Exportação de um ElastiCache backup (console)

As etapas a seguir usam o ElastiCache console para exportar um backup para um bucket do Amazon S3 para que você possa acessá-lo de fora. ElastiCache O bucket do Amazon S3 deve estar na mesma AWS região do ElastiCache backup.

Para exportar um ElastiCache backup para um bucket do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista dos seus backups, no painel de navegação esquerdo, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup que você deseja exportar.
4. Escolha Copiar.
5. Em Create a Copy of the Backup? (Criar uma cópia do backup?), faça o seguinte:
 - a. Na caixa New backup name (Novo nome de backup), digite um nome para seu novo backup.

O nome deve ter entre 1 e 1.000 caracteres e pode ser codificado em UTF-8.

ElastiCache adiciona um identificador de instância e `.rdb` ao valor que você insere aqui. Por exemplo, se você inserir `my-exported-backup`, ElastiCache criará `my-exported-backup-0001.rdb`.

- b. Na lista Target S3 Location (Local do S3 de destino), escolha o nome do bucket do Amazon S3 para o qual você deseja copiar seu backup (o bucket que você criou em [Crie um bucket Amazon S3](#)).

O local de destino do S3 deve ser um bucket do Amazon S3 na região AWS do backup com as seguintes permissões para que o processo de exportação seja bem-sucedido.

- Acesso ao objeto: Ler e Escrever.
- Permissões de acesso: Ler.

Para obter mais informações, consulte [Conceda ElastiCache acesso ao seu bucket Amazon S3](#).

- c. Escolha Copiar.

Note

Se seu bucket do S3 não tiver as permissões necessárias ElastiCache para exportar um backup para ele, você receberá uma das seguintes mensagens de erro. Retorne para [Conceda ElastiCache acesso ao seu bucket Amazon S3](#) a fim de adicionar as permissões especificadas e tente exportar o backup novamente.

- ElastiCache não recebeu permissões READ %s no S3 Bucket.

Solução: adicione permissões de Leitura no bucket.

- ElastiCache não recebeu permissões de GRAVAÇÃO %s no S3 Bucket.

Solução: adicione permissões de Gravação no bucket.

- ElastiCache não recebeu as permissões READ_ACP %s no S3 Bucket.

Solução: adicione permissão de acesso de Leitura no bucket.

Se você quiser copiar seu backup para outra AWS região, use o Amazon S3 para copiá-lo. Para obter mais informações, consulte [Cópia de objetos](#) no Guia do usuário do Amazon Simple Storage Service.

Exportação de um ElastiCache backup sem servidor (AWS CLI)

Exportação de um backup de um cache sem servidor

Exporte o backup para um bucket do Amazon S3 usando a operação `export-serverless-cache-snapshot` da CLI com os seguintes parâmetros:

Parâmetros

- `--serverless-cache-snapshot-name` - Nome do backup a ser copiado.
- `--s3-bucket-name`: nome do bucket do Amazon S3 no qual você deseja exportar o backup. Uma cópia do backup é feita no bucket especificado.
 - `--s3-bucket-name` Deve ser um bucket do Amazon S3 na AWS região do backup com as seguintes permissões para que o processo de exportação seja bem-sucedido.
 - Acesso ao objeto: Ler e Escrever.
 - Permissões de acesso: Ler.

A operação a seguir copia um backup para `my-s3-bucket`.

Para Linux, macOS ou Unix:

```
aws elasticache export-serverless-cache-snapshot \
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \
  --s3-bucket-name my-s3-bucket
```

Para Windows:

```
aws elasticache export-serverless-cache-snapshot ^
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^
  --s3-bucket-name my-s3-bucket
```

Exportando um backup de ElastiCache cluster baseado em nós (AWS CLI)

Exportar um backup de um cluster baseado em nós

Exporte o backup para um bucket do Amazon S3 usando a operação `copy-snapshot` da CLI com os seguintes parâmetros:

Parâmetros

- `--source-snapshot-name` - Nome do backup a ser copiado.
- `--target-snapshot-name` - Nome da cópia do backup.

O nome deve ter entre 1 e 1.000 caracteres e pode ser codificado em UTF-8.

ElastiCache adiciona um identificador de instância e `.rdb` ao valor inserido aqui. Por exemplo, se você inserir `my-exported-backup`, ElastiCache criará `my-exported-backup-0001.rdb`.

- `--target-bucket`: nome do bucket do Amazon S3 no qual você deseja exportar o backup. Uma cópia do backup é feita no bucket especificado.

`--target-bucket` Deve ser um bucket do Amazon S3 na AWS região do backup com as seguintes permissões para que o processo de exportação seja bem-sucedido.

- Acesso ao objeto: Ler e Escrever.
- Permissões de acesso: Ler.

Para obter mais informações, consulte [Conceda ElastiCache acesso ao seu bucket Amazon S3](#).

A operação a seguir copia um backup para `my-s3-bucket`.

Para Linux, macOS ou Unix:

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 \  
  --target-snapshot-name my-exported-backup \  
  --target-bucket my-s3-bucket
```

Para Windows:

```
aws elasticache copy-snapshot ^  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 ^  
  --target-snapshot-name my-exported-backup ^  
  --target-bucket my-s3-bucket
```

Restauração de um backup para um novo cache

Você pode restaurar um backup existente do Valkey em um novo cache do Valkey ou cluster baseado em nós e restaurar um backup existente do Redis OSS em um novo cache do Redis OSS ou cluster baseado em nós. Você também pode restaurar um backup existente de cache de tecnologia sem servidor do Memcached em um novo cache de tecnologia sem servidor do Memcached.

Restauração de um backup para um cache sem servidor (console)

Note

ElastiCache O Serverless suporta arquivos RDB compatíveis com Valkey 7.2 e superior e versões do Redis OSS entre 5.0 e a versão mais recente disponível.

Para restaurar um backup para um cache sem servidor (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup que você deseja restaurar.
4. Escolha Ações e Restaurar
5. Insira um nome para o novo cache sem servidor e uma descrição opcional.
6. Clique em Criar para criar o novo cache e importar dados do backup.

Restauração de um backup para um cluster baseado em nós (console)

Para restaurar um backup para um cluster baseado em nós (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup do qual você deseja restaurar.
4. Escolha Ações e Restaurar

5. Escolha Cache baseado em nós e personalize as configurações do cluster, como tipo de nó, tamanhos, número de fragmentos, réplicas, posicionamento AZ e configurações de segurança.
6. Escolha Criar para criar o novo cluster baseado em nós e importar dados do backup.

Restauração de um backup para um cache sem servidor (AWS CLI)

Note

ElastiCache O Serverless suporta arquivos RDB compatíveis com Valkey 7.2 e superior e versões do Redis OSS entre 5.0 e a versão mais recente disponível.

Para restaurar um backup para um cache sem servidor novo (AWS CLI)

O AWS CLI exemplo a seguir cria um novo cache usando `create-serverless-cache` e importa dados de um backup.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Exclusão de um backup

Um backup automático é excluído automaticamente quando o limite de retenção expira. Se você excluir um cluster, todos os seus backups automáticos também serão excluídos. Se você excluir um grupo de replicação, todos os backups automáticos dos clusters desse grupo também serão excluídos.

ElastiCache fornece uma operação de API de exclusão que permite excluir um backup a qualquer momento, independentemente de o backup ter sido criado automática ou manualmente. Como os backups manuais não possuem um limite de retenção, a exclusão manual é a única maneira de removê-los.

Você pode excluir um backup usando o ElastiCache console AWS CLI, o ou a ElastiCache API.

Exclusão de um backup (console)

O procedimento a seguir exclui um backup usando o ElastiCache console.

Para excluir um backup

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione Backups.

A tela Backups aparece com uma lista dos seus backups.

3. Escolha a caixa à esquerda do nome do backup que você deseja excluir.
4. Escolha Excluir.
5. Se você quiser excluir esse backup, escolha Excluir na tela de confirmação Excluir backup. O status muda para deleting.

Exclusão de um backup sem servidor (AWS CLI)

Use a AWS CLI operação delete-snapshot com o parâmetro a seguir para excluir um backup sem servidor.

- `--serverless-cache-snapshot-name` - Nome do backup a ser excluído.

O código a seguir exclui o backup myBackup.

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-name myBackup
```

Para obter mais informações, consulte [delete-serverless-cache-snapshot](#) na Referência de comandos da AWS CLI.

Excluir um backup de cluster baseado em nós (AWS CLI)

Use a AWS CLI operação `delete-snapshot` com o parâmetro a seguir para excluir um backup de cluster baseado em nós.

- `--snapshot-name` - Nome do backup a ser excluído.

O código a seguir exclui o backup `myBackup`.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Para obter mais informações, consulte [delete-snapshot](#) na Referência de comandos da AWS CLI.

Marcação de backups

Você pode atribuir os próprios metadados a cada backup na forma de tags. As tags permitem categorizar seus backups de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando você tem muitos recursos do mesmo tipo. É possível identificar rapidamente um recurso específico baseado nas tags que você atribuiu a ele. Para obter mais informações, consulte [Recursos que podem ser marcados](#).

Tags de alocação de custos são um meio de rastrear seus custos em vários serviços da AWS, agrupando suas despesas em faturas por valores de tag. Para saber mais sobre alocação de custos, consulte [Usar tags de alocação de custos](#).

Usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache, você pode adicionar, listar, modificar, remover ou copiar tags de alocação de custos em seus backups. Para obter mais informações, consulte [Monitoramento de custos com tags de alocação de custos](#).

Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente

Ao criar um novo cluster baseado em nós do Valkey ou Redis OSS, você pode propagá-lo com dados de um arquivo de backup .rdb do Valkey ou Redis OSS. A semeadura do cluster é útil se você atualmente gerencia uma instância Valkey ou Redis OSS fora ElastiCache e deseja preencher seu novo ElastiCache cluster baseado em nós do Redis OSS com seus dados existentes do Valkey ou do Redis OSS.

Para implantar um novo cluster baseado em nós Valkey ou Redis OSS a partir de um backup Valkey ou Redis OSS criado na Amazon, consulte ElastiCache [Restauração de um backup para um novo cache](#)

Ao usar um arquivo .rdb do Valkey ou Redis OSS para propagar um novo cluster baseado em nós, você pode fazer o seguinte:

- Atualização de um cluster não particionado para um cluster baseado em nós do Valkey ou Redis OSS (modo cluster habilitado) que execute o Redis OSS versão 3.2.4.
- Especifique um número de fragmentos (chamados de grupos de nós na API ou na CLI) no novo cluster baseado em nós. Esse número pode ser diferente do número de fragmentos no cluster baseado em nós que foi usado para criar o arquivo de backup.
- Especificar um tipo de nó diferente para o novo cluster baseado em nós, maior ou menor que o usado no cluster que fez o backup. Se você escalar para um tipo de nó menor, certifique-se de que o novo tipo de nó tenha memória suficiente para seus dados e a sobrecarga do Valkey ou Redis OSS. Para obter mais informações, consulte [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#).
- Distribua suas chaves nos slots do novo cluster do Valkey ou Redis OSS (modo cluster habilitado) de forma diferente do que no cluster que foi usado para criar o arquivo de backup.

Note

Não é possível propagar um cluster do Valkey ou Redis OSS (modo cluster desabilitado) a partir de um arquivo .rdb criado a partir de um cluster do Valkey ou Redis OSS (modo cluster habilitado).

⚠ Important

- É necessário garantir que seus dados de backup do Valkey ou Redis OSS não excedam os recursos do nó. Por exemplo, você não pode fazer upload de um arquivo .rdb com 5 GB de dados do Valkey ou Redis OSS em um nó cache.m3.medium que tenha 2,9 GB de memória.

Se o backup for muito grande, o cluster resultante terá um status de `restore-failed`. Se isso acontecer, você deverá excluir o cluster e começar de novo.

Para obter uma lista completa dos tipos e especificações de nós, consulte os [recursos Parâmetros específicos de node-type do Redis OSS e detalhes dos ElastiCache produtos da Amazon](#).

- Só é possível criptografar um arquivo .rdb do Valkey ou Redis OSS com a criptografia do lado do servidor do Amazon S3 (SSE-S3). Para obter mais informações, consulte [Proteger dados usando a criptografia no lado do servidor](#).

A seguir, você encontrará tópicos que orientam você na migração do seu cluster de fora do Valkey ou do Redis OSS ElastiCache para o Redis OSS. ElastiCache

Migrando ElastiCache para o Redis OSS

- [Etapa 1: criar um backup do Valkey ou Redis OSS](#)
- [Etapa 2: criar um bucket e uma pasta no Amazon S3](#)
- [Etapa 3: carregar seu backup no Amazon S3](#)
- [Etapa 4: conceder acesso de ElastiCache leitura ao arquivo.rdb](#)

Migração de serviços externos ElastiCache para o Redis OSS.

- [Etapa 1: criar um backup do Valkey ou Redis OSS](#)
- [Etapa 2: criar um bucket e uma pasta no Amazon S3](#)
- [Etapa 3: carregar seu backup no Amazon S3](#)
- [Etapa 4: conceder acesso de ElastiCache leitura ao arquivo.rdb](#)

Etapa 1: criar um backup do Valkey ou Redis OSS

Para criar o backup do Valkey ou do Redis OSS para alimentar sua ElastiCache instância do Redis OSS

1. Conecte-se à sua instância do Valkey ou Redis OSS existente.
2. Execute a operação BGSAVE ou SAVE para criar um backup. Observe onde seu arquivo .rdb está localizado.

BGSAVE é assíncrono e não bloqueia outros clientes durante o processamento. Para obter mais informações, consulte [BGSAVE](#) no site do Valkey.

SAVE é síncrono e bloqueia outros processos até terminar. Para obter mais informações, consulte [SAVE](#) no site do Valkey.

Para obter informações adicionais sobre como criar um backup, consulte [Persistência](#) no site do Valkey.

Etapa 2: criar um bucket e uma pasta no Amazon S3

Quando você tiver criado o arquivo de backup, precisará carregá-lo em uma pasta dentro de um bucket do Amazon S3. Para fazer isso, primeiro você deve ter um bucket do Amazon S3 e uma pasta dentro desse bucket. Se você já possui um bucket do Amazon S3 e uma pasta com as permissões apropriadas, poderá pular para [Etapa 3: carregar seu backup no Amazon S3](#).

Como criar um bucket do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Para criar um bucket do Amazon S3, siga as instruções em [Criação de um bucket](#) no Guia do usuário do Amazon Simple Storage Service.

O nome do bucket do Amazon S3 deve estar em conformidade com o DNS. Caso contrário, não ElastiCache conseguirá acessar seu arquivo de backup. As regras para a conformidade de DNS são:

- Os nomes devem ter no mínimo 3 e no máximo 63 caracteres de extensão.
- Os nomes devem ser uma série de um ou mais rótulos separados por um ponto (.) em que cada rótulo:

- Começa com uma letra minúscula ou um número.
- Termina com uma letra minúscula ou um número.
- Contém somente letras minúsculas, números e traços.
- Os nomes não podem ser formatado como um endereço IP (por exemplo, 192.0.2.0).

Você deve criar seu bucket Amazon S3 na mesma AWS região do seu novo cluster OSS ElastiCache para Redis. Essa abordagem garante a maior velocidade de transferência de dados ao ElastiCache ler seu arquivo.rdb do Amazon S3.

 Note

Para manter seus dados da forma mais segura possível, restrinja ao máximo as permissões em seu bucket do Amazon S3. Ao mesmo tempo, as permissões ainda precisam permitir que o bucket e seu conteúdo seja usado para propagar o novo cluster do Valkey ou Redis OSS.

Para adicionar uma pasta a um bucket do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Escolha o nome do bucket para o qual deseja fazer upload do arquivo .rdb.
3. Selecione Criar pasta.
4. Insira um nome para a nova pasta.
5. Escolha Salvar.

Anote o nome do bucket e o nome da pasta.

Etapa 3: carregar seu backup no Amazon S3

Agora, faça upload do arquivo .rdb criado em [Etapa 1: criar um backup do Valkey ou Redis OSS](#). Carregue-o no bucket e na pasta do Amazon S3 que você criou em [Etapa 2: criar um bucket e uma pasta no Amazon S3](#). Para obter mais informações sobre essa tarefa, consulte [Adicionar um objeto a um bucket](#). Entre as etapas 2 e 3, escolha o nome da pasta que você criou.

Para carregar seu arquivo .rdb em uma pasta do Amazon S3

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Escolha o nome do bucket do Amazon S3 criado na Etapa 2.
3. Escolha o nome da pasta que você criou na Etapa 2.
4. Escolha Carregar.
5. Escolha Adicionar arquivos.
6. Navegue para encontrar um ou mais arquivos que deseja carregar e depois escolha esses arquivos. Para escolher vários arquivos, mantenha pressionada a tecla Ctrl enquanto escolhe o nome de cada arquivo.
7. Escolha Open (Abrir).
8. Confirme se os arquivos corretos estão listados na caixa de diálogo Upload e depois escolha Upload.

Anote o caminho para o arquivo .rdb. Por exemplo, se o nome do bucket for myBucket e o caminho for myFolder/redis.rdb, insira myBucket/myFolder/redis.rdb. Você precisa desse caminho para propagar o novo cluster com os dados neste backup.

Para obter mais informações, consulte [Restrições e limitações de bucket](#) no Guia do usuário do Amazon Simple Storage Service.

Etapa 4: conceder acesso de ElastiCache leitura ao arquivo.rdb

Agora, conceda acesso de ElastiCache leitura ao seu arquivo de backup .rdb. Você concede ElastiCache acesso ao seu arquivo de backup de uma forma diferente, dependendo se seu bucket está em uma AWS região padrão ou em uma AWS região opcional.

AWS As regiões introduzidas antes de 20 de março de 2019 estão habilitadas por padrão. Você pode começar a trabalhar nessas AWS regiões imediatamente. As regiões introduzidas após 20 de março de 2019, como Ásia-Pacífico (Hong Kong) e Oriente Médio (Bahrein), são desabilitadas por padrão. Você deve habilitar ou escolher essas regiões para poder usá-las, conforme descrito em [Gerenciar regiões da AWS](#) na Referência geral da AWS.

Escolha sua abordagem de acordo com sua AWS região:

- Para uma região padrão, use o procedimento em [Conceder acesso de ElastiCache leitura ao arquivo.rdb em uma região padrão](#).

- Para uma região opcional, use o procedimento em [Conceda acesso de ElastiCache leitura ao arquivo.rdb em uma região opcional](#).

Conceder acesso de ElastiCache leitura ao arquivo.rdb em uma região padrão

AWS As regiões introduzidas antes de 20 de março de 2019 estão habilitadas por padrão. Você pode começar a trabalhar nessas AWS regiões imediatamente. As regiões introduzidas após 20 de março de 2019, como Ásia-Pacífico (Hong Kong) e Oriente Médio (Bahrein), são desabilitadas por padrão. Você deve habilitar ou escolher essas regiões para poder usá-las, conforme descrito em [Gerenciar regiões da AWS](#) na Referência geral da AWS.

Para conceder acesso de ElastiCache leitura ao arquivo de backup em uma AWS região ativada por padrão

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Escolha o nome do bucket do S3 que contém seu arquivo .rdb.
3. Escolha o nome da pasta que contém seu arquivo .rdb.
4. Escolha o nome do seu arquivo de backup .rdb. O nome do arquivo selecionado aparece acima das guias na parte superior da página.
5. Escolha Permissões.
6. Se aws-scs-ssomente 3 leituras ou uma das canônicas IDs na lista a seguir não estiver listada como usuário, faça o seguinte:
 - a. Em Acesso para outras AWS contas, escolha Adicionar favorecido.
 - b. Na caixa, adicione a ID canônica da AWS região conforme mostrado a seguir:

- AWS GovCloud Região (Oeste dos EUA):

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

Important

O backup deve estar localizado em um bucket S3 AWS GovCloud (US) para que você possa baixá-lo em um cluster Valkey ou Redis OSS em AWS GovCloud (US)

- AWS Regiões ativadas por padrão:

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

- c. Defina as permissões no bucket escolhendo Yes (Sim) para o seguinte:
 - Objeto de lista/gravação
 - Permissões ACL de objetos de leitura/gravação
 - d. Escolha Salvar.
7. Escolha Overview (Visão geral) e escolha Download (Fazer download).

Conceda acesso de ElastiCache leitura ao arquivo.rdb em uma região opcional

AWS As regiões introduzidas antes de 20 de março de 2019 estão habilitadas por padrão. Você pode começar a trabalhar nessas AWS regiões imediatamente. As regiões introduzidas após 20 de março de 2019, como Ásia-Pacífico (Hong Kong) e Oriente Médio (Bahrein), são desabilitadas por padrão. Você deve habilitar ou escolher essas regiões para poder usá-las, conforme descrito em [Gerenciar regiões da AWS](#) na Referência geral da AWS.

Agora, conceda acesso de ElastiCache leitura ao seu arquivo de backup .rdb.

Para conceder acesso de ElastiCache leitura ao arquivo de backup

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. <https://console.aws.amazon.com/s3/>
2. Escolha o nome do bucket do S3 que contém seu arquivo .rdb.
3. Escolha o nome da pasta que contém seu arquivo .rdb.
4. Escolha o nome do seu arquivo de backup .rdb. O nome do arquivo selecionado aparece acima das guias na parte superior da página.
5. Escolha a guia Permissões.
6. Em Permissões, escolha Política de bucket e, em seguida, Editar.
7. Atualize a política para conceder as permissões ElastiCache necessárias para realizar operações:
 - Adicione ["Service" : "*region-full-name*.elasticache-snapshot.amazonaws.com"] a Principal.

- Adicione as seguintes permissões necessárias para exportar um snapshot para o bucket do Amazon S3:
 - "s3:GetObject"
 - "s3:ListBucket"
 - "s3:GetBucketAcl"

Veja a seguir um exemplo de como a política atualizada pode parecer.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/backup1.rdb",
        "arn:aws:s3:::amzn-s3-demo-bucket/backup2.rdb"
      ]
    }
  ]
}
```

8. Escolha Salvar alterações.

Semeie o ElastiCache cluster com os dados do arquivo.rdb

Agora você está pronto para criar um ElastiCache cluster e plantá-lo com os dados do arquivo.rdb. Para criar o cluster, siga as instruções em [Criação de um cluster do Valkey ou Redis OSS](#) ou [Criação de um grupo de replicação do Valkey ou Redis OSS do início](#). Certifique-se de escolher o Valkey ou Redis OSS como seu mecanismo de cluster.

O método usado para saber ElastiCache onde encontrar o backup que você enviou para o Amazon S3 depende do método usado para criar o cluster:

Prepare o ElastiCache cluster ou grupo de replicação do for Redis OSS com os dados do arquivo.rdb

- Usando o ElastiCache console

Ao selecionar Cluster settings (Configurações de cluster), escolha Restore from backups (Restaurar de backups) como seu método de criação de cluster e escolha Other backups (Outros backups) como sua Source (Origem) na seção Backup source (Origem de backup). Na caixa Propagar local S3 do arquivo RDB, digite o caminho do Amazon S3 para o(s) arquivo(s). Se você tiver vários arquivos .rdb, digite o caminho para cada um em uma lista separada por vírgulas. O caminho do Amazon S3 parece-se com *myBucket/myFolder/myBackupFilename*.rdb.

- Usando o AWS CLI

Se você usar a operação create-cache-cluster ou create-replication-group, use o parâmetro --snapshot-arns para especificar um ARN totalmente qualificado para cada arquivo .rdb. Por exemplo, .arn:aws:s3:::*myBucket/myFolder/myBackupFilename*.rdb O ARN deve ser resolvido para os arquivos de backup que você armazenou no Amazon S3.

- Usando a ElastiCache API

Se você usar a operação CreateCacheCluster ou a CreateReplicationGroup ElastiCache API, use o parâmetro SnapshotArns para especificar um ARN totalmente qualificado para cada arquivo.rdb. Por exemplo, .arn:aws:s3:::*myBucket/myFolder/myBackupFilename*.rdb O ARN deve ser resolvido para os arquivos de backup que você armazenou no Amazon S3.

Important

Ao propagar um cluster do Valkey ou Redis OSS (modo cluster habilitado), você precisa configurar cada grupo de nós (fragmento) no novo cluster ou grupo de replicação. Use o parâmetro --node-group-configuration (API: NodeGroupConfiguration) para fazer isso. Para saber mais, consulte:

- CLI: [create-replication-group](#) na referência AWS CLI
- API: [CreateReplicationGroup](#) na referência da ElastiCache API

Durante o processo de criação do cluster, os dados no backup do Valkey ou Redis OSS são gravados no cluster. Você pode monitorar o progresso visualizando as mensagens do ElastiCache evento. Para fazer isso, consulte o ElastiCache console e escolha Eventos de cache. Você também pode usar a interface de linha de AWS ElastiCache comando ou a ElastiCache API para obter mensagens de eventos. Para obter mais informações, consulte [Visualizando ElastiCache eventos](#).

Versões e atualizações do motor em ElastiCache

Esta seção aborda os mecanismos Valkey, Memcached e Redis OSS aceitos e como realizar atualizações. Observe que todos os atributos disponíveis com o Redis OSS 7.2 estão disponíveis no Valkey 7.2 e posteriores, por padrão. Você também pode fazer o upgrade de alguns mecanismos existentes ElastiCache do Redis OSS para um mecanismo Valkey.

Atualização das versões do mecanismo, incluindo atualizações de mecanismos cruzados

Valkey e Redis OSS

Com Valkey e Redis OSS, você inicia atualizações de versão no seu cluster ou grupo de replicação modificando-o com o uso do console do ElastiCache, da AWS CLI ou da API do ElastiCache e especificando uma versão mais nova do mecanismo.

Você também pode fazer uma atualização cruzada do Redis OSS para o Valkey. Para obter mais informações sobre como fazer atualizações cruzadas, consulte [Como atualizar do Redis OSS para o Valkey](#).

Tópicos

- [Como atualizar do Redis OSS para o Valkey](#)
- [Resolver bloqueios de atualização do mecanismo Valkey ou Redis OSS](#)

Como modificar clusters e grupos de replicação	
Caches	Grupos de replicação
Uso do Console de gerenciamento da AWS do ElastiCache	Usando o Console de gerenciamento da AWS
Uso da AWS CLI com o ElastiCache	Usando o AWS CLI
Uso da API do ElastiCache	Usando a ElastiCache API

Memcached

Com o Memcached, para iniciar as atualizações de versão no seu cluster, modifique-o e especifique uma versão mais recente do mecanismo. Faça isso usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache:

- Para usar a Console de gerenciamento da AWS, consulte - [Uso do Console de gerenciamento da AWS do ElastiCache](#).
- Para usar a AWS CLI, consulte [Uso da AWS CLI com o ElastiCache](#).
- Para usar a API do ElastiCache, consulte [Uso da API do ElastiCache](#).

Como atualizar do Redis OSS para o Valkey

O Valkey foi projetado como um substituto imediato para o Redis OSS 7. Você pode atualizar do Redis OSS para o Valkey usando o Console, a API ou a CLI, especificando o novo mecanismo e a versão principal do mecanismo. O endereço IP do endpoint e todos os outros aspectos do aplicativo não serão alterados pela atualização. Ao atualizar do Redis OSS 5.0.6 e superior, você não terá tempo de inatividade.

Note

Requisitos da versão CLI da AWS para atualizações do Redis OSS para Valkey:

- Para CLI da AWS v1: versão mínima exigida 1.35.2 (versão atual: 1.40.22)
- Para CLI da AWS v2: versão mínima exigida 2.18.2 (versão atual: 2.27.22)

Note

- Ao atualizar de versões anteriores do Redis OSS 5.0.6, você pode experimentar um tempo de failover de 30 a 60 segundos durante a propagação do DNS.
- Para atualizar um cluster de nó único Redis OSS (modo cluster desabilitado) existente para o mecanismo Valkey, primeiro siga estas etapas: [Criação de um grupo de replicação usando um cluster existente](#). Depois que o cluster de nó único Redis OSS (modo cluster desabilitado) tiver sido adicionado a um grupo de replicação, você pode fazer atualização cruzada entre mecanismos para o Valkey.

Atualização de um grupo de replicação do Redis OSS para o Valkey

Se você tiver um grupo de replicação Redis OSS existente que esteja usando o grupo de parâmetros de cache padrão, poderá atualizar para o Valkey especificando o novo mecanismo e a versão do mecanismo com a API `modify-replication-group`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --engine valkey \  
  --engine-version 8.0
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --engine valkey ^  
  --engine-version 8.0
```

Se você tiver um grupo de parâmetros de cache personalizado aplicado ao grupo de replicação do Redis OSS existente que deseja atualizar, também será necessário passar um grupo de parâmetros de cache personalizado do Valkey na solicitação. O grupo de parâmetros personalizados de entrada do Valkey deve ter os mesmos valores de parâmetros estáticos do Redis OSS que o grupo de parâmetros personalizados do Redis OSS existente.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --engine valkey \  
  --engine-version 8.0 \  
  --cache-parameter-group-name myParamGroup
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --engine valkey ^  
  --engine-version 8.0 ^
```

```
--cache-parameter-group-name myParamGroup
```

Atualização de um cache de tecnologia sem servidor Redis OSS para Valkey com a CLI

Para Linux, macOS ou Unix:

```
aws elasticache modify-serverless-cache \  
  --serverless-cache-name myCluster \  
  --engine valkey \  
  --major-engine-version 8
```

Para Windows:

```
aws elasticache modify-serverless-cache ^  
  --serverless-cache-name myCluster ^  
  --engine valkey ^  
  --major-engine-version 8
```

Atualização do Redis OSS para o Valkey com o Console

Atualização do Redis OSS 5 para o Valkey

1. Selecione o cache do Redis OSS para atualizar.
2. Uma janela Atualizar para o Valkey deve aparecer. Selecione o botão Atualizar para o Valkey.
3. Acesse Configurações de cache e selecione Versão do mecanismo. A versão mais recente do Valkey é recomendada.
4. Se esse cache for de tecnologia sem servidor, você precisará atualizar o grupo de parâmetros. Vá para a área Grupos de parâmetros das Configurações de cache e selecione um grupo de parâmetros apropriado, como default.valkey8.
5. Selecione Atualizar.

Esse cache agora será listado na área Valkey do console.

Note

A atualização diretamente do Redis OSS 4 ou inferior para o Valkey pode incluir um tempo de failover maior de 30 a 60 segundos durante a propagação do DNS.

Como fazer o downgrade do Valkey para o Redis OSS

Se, por algum motivo, você desejar reverter seu cluster atualizado, o Amazon ElastiCache é compatível com a reversão de um cache do Valkey 7.2 para o Redis OSS 7.1. Você pode realizar uma reversão usando as mesmas etapas de console, API ou CLI de uma atualização do mecanismo e especificando o Redis OSS 7.1 como a versão do mecanismo de destino. As reversões usam os mesmos processos de uma atualização. O endereço IP do endpoint e todos os outros aspectos do aplicativo não serão alterados pela reversão e você não terá tempo de inatividade.

Além disso, você pode restaurar um instantâneo criado a partir do cache do Valkey 7.2 como um cache do Redis OSS 7.1. Ao restaurar a partir de um snapshot, você pode especificar o Redis OSS 7.1 como a versão do mecanismo de destino. Ao usar essa opção, um novo cache será criado a partir do instantâneo. A restauração a partir de um instantâneo não tem efeito no cache do Valkey a partir do qual o instantâneo foi criado.

Os seguintes requisitos e limitações se aplicam ao realizar uma reversão:

- O ElastiCache só é compatível com a reversão do Valkey 7.2 para o Redis OSS 7.1. Isso é verdade mesmo se você tiver atualizado para o Valkey 7.2 de uma versão anterior ao Redis OSS 7.1.
- Qualquer grupo de usuários e usuários associados ao grupo de replicação ou ao cache com tecnologia sem servidor que está sendo revertido devem ser configurados com o tipo de mecanismo REDIS.

Resolver bloqueios de atualização do mecanismo Valkey ou Redis OSS

Conforme mostrado na tabela a seguir, a operação de atualização do mecanismo Valkey ou Redis OSS será bloqueada se você tiver uma operação pendente de aumento vertical da escala.

Operações pendentes	Operações bloqueadas
Amplie a sua capacidade	Atualização imediata do mecanismo
Atualização do mecanismo	Expansão imediata
Expansão e atualização do mecanismo	Expansão imediata
	Atualização imediata do mecanismo

Para resolver uma atualização bloqueada do mecanismo do Redis OSS

- Execute um destes procedimentos:

- Planeje sua operação de atualização do mecanismo Redis OSS ou Valkey para a próxima janela de manutenção, desmarcando a caixa de seleção Apply immediately (Aplicar imediatamente).

Com a CLI, use `--no-apply-immediately`. Com a API, use `ApplyImmediately=false`.

- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de atualização do mecanismo Redis OSS.
- Adicione a operação de aumento vertical da escala do Redis OSS a essa modificação de cluster com a caixa de seleção Apply Immediately (Aplicar imediatamente) marcada.

Com a CLI, use `--apply-immediately`. Com a API, use `ApplyImmediately=true`.

Essa abordagem cancela efetivamente a atualização do mecanismo durante a próxima janela de manutenção, executando-a imediatamente.

Suporte estendido do ElastiCache

Com o suporte estendido do ElastiCache, é possível continuar executando o cache em uma versão principal do mecanismo após a data de término do suporte padrão por um custo adicional. Se você não fizer o upgrade após o término da data de suporte padrão, você será cobrado.

O Suporte estendido fornece as seguintes atualizações e suporte técnico:

- Atualizações de segurança para CVEs críticos e altos para seu cache e mecanismo de cache
- Correções de erros e patches para problemas críticos
- A capacidade de abrir casos de suporte e receber ajuda para solução de problemas dentro do acordo de serviço padrão do ElastiCache

Essa oferta paga oferece a você mais tempo para atualizar para uma versão principal compatível do mecanismo.

Por exemplo, a data de término do suporte padrão do ElastiCache para o Redis OSS 4.0.10 é 31 de janeiro de 2026. Se você não estiver pronto para atualizar manualmente para o Valkey ou para o Redis OSS 6 ou posterior até essa data, o ElastiCache inscreverá automaticamente seus

caches no Suporte estendido e você poderá continuar executando o Redis OSS 4.0.10. A partir do primeiro dia do mês após o término do suporte padrão, 1º de fevereiro de 2026, o ElastiCache cobra automaticamente pelo Suporte estendido.

O Suporte estendido para uma versão principal do mecanismo ficará disponível até três anos após a data de término do suporte padrão. Para as versões 4 e 5 do Elasticache para Redis OSS, isso será em 31 de janeiro de 2029. Após essa data, todos os caches que ainda executam as versões 4 e 5 do Redis OSS serão automaticamente atualizados para a versão mais recente do Valkey.

Quando o período de suporte de um mecanismo terminar, os caches que continuarem executando essa versão antiga passarão automaticamente para o Suporte estendido. Você será notificado antes da data de início dos preços do Suporte estendido para que, em vez disso, você possa atualizar sua instância. Você também pode cancelar explicitamente a qualquer momento fazendo o upgrade para as versões compatíveis.

Para ter mais informações sobre as datas de fim do suporte padrão e do Suporte estendido, consulte [ElastiCache versões para o cronograma de fim de vida útil do Redis OSS](#) para Valkey, Memcached ou Redis OSS.

Tópicos

- [Cobranças do Suporte estendido do ElastiCache](#)
- [Versões com o Suporte estendido do ElastiCache](#)
- [ElastiCache e responsabilidades do cliente com o Suporte estendido do ElastiCache](#)

Cobranças do Suporte estendido do ElastiCache

Você pagará por todos os mecanismos inscritos no Suporte estendido do ElastiCache desde o dia seguinte à data de término do suporte padrão. Para saber a data de término do suporte padrão do ElastiCache, consulte [Versões com o Suporte estendido do ElastiCache](#).

A cobrança adicional pelo Suporte estendido do ElastiCache é interrompida automaticamente ao realizar uma das seguintes ações:

- Atualizar para uma versão do mecanismo coberta pelo suporte padrão.
- Excluir o cache que está executando uma versão principal após a data de fim do suporte padrão do ElastiCache.

As cobranças serão reiniciadas se a versão do mecanismo de destino entrar no Suporte estendido no futuro.

Por exemplo, digamos que o ElastiCache versão 4 para Redis OSS entre no Suporte estendido em 1º de fevereiro de 2026 e você atualize seus caches da v4 para a v6 em 1º de janeiro de 2027. Você só será cobrado por 11 meses de Suporte estendido, no ElastiCache versão 4 para Redis OSS. Se você continuar executando o ElastiCache versão 6 para Redis OSS após a data de fim do suporte padrão de 31 de janeiro de 2027, esses caches voltarão a incorrer em cobranças de Suporte estendido a partir de 1º de fevereiro de 2027.

Você pode evitar a cobrança pelo Suporte do ElastiCacheExtended impedindo que o ElastiCache crie ou restaure um cache após a data de fim do suporte padrão do ElastiCache.

Para obter mais informações, consulte [Definição de preço Amazon ElastiCache](#).

Versões com o Suporte estendido do ElastiCache

As versões 4 e 5 do Redis Open Source Software (OSS) atingiram o fim da vida útil da comunidade em 2020 e 2022, respectivamente. Isso significa que a comunidade não lançará mais atualizações, correções de erros ou patches de segurança. O suporte padrão para as versões 4 e 5 do ElastiCache Redis OSS no ElastiCache terminará em 31 de janeiro de 2026. Continuar usando versões não suportadas do Redis OSS pode deixar seus dados vulneráveis a [vulnerabilidades e exposições comuns](#) (CVEs) conhecidas.

A partir de 1º de fevereiro de 2026, os caches do ElastiCache ainda em execução nas versões 4 e 5 do Redis OSS serão automaticamente registrados no Suporte estendido, para fornecer disponibilidade e segurança contínuas. Embora o Suporte estendido ofereça flexibilidade, recomendamos tratar o fim do suporte padrão como um marco de planejamento para suas workloads de produção. Recomendamos fortemente que você atualize seus caches do Redis OSS v4 e v5 para o ElastiCache for Valkey ou o Redis OSS v6 ou posterior, antes do fim do suporte padrão.

A tabela a seguir resume a data de fim do suporte padrão e as datas de Suporte estendido do Amazon ElastiCache.

Suporte estendido e cronograma de fim de vida

Versão principal do mecanismo	Fim do suporte padrão	Início do Suporte estendido premium do ano 1	Início do Suporte estendido premium do ano 2	Início do Suporte estendido premium do ano 3	Fim do Suporte estendido e versão EOL
Redis OSS v4	31/1/2026	1/2/2026	1/2/2027	1/2/2028	1/31/2029
Redis OSS v5	31/1/2026	1/2/2026	1/2/2027	1/2/2028	1/31/2029
Redis OSS v6	1/31/2027	1/2/2027	1/2/2028	1/2/2029	1/31/2030

O Suporte estendido só será oferecido para a versão de patch mais recente compatível com cada versão principal do Redis OSS. Quando o Suporte estendido começar em 1º de fevereiro de 2026, se seus clusters do Redis OSS v4 e v5 ainda não estiverem nas versões de patch mais recentes, eles serão automaticamente atualizados para v4.0.10 para Redis OSS v4 e v5.0.6 para Redis OSS v5, antes de serem inscritos no Suporte estendido. Isso garante que você receba atualizações de segurança e correções de bugs por meio do Suporte estendido. Você não precisa realizar nenhuma ação para atualizar para essas versões de patch mais recentes como parte da transição do Suporte estendido.

ElastiCache e responsabilidades do cliente com o Suporte estendido do ElastiCache

A seguir estão as responsabilidades do Amazon ElastiCache e suas responsabilidades com o Suporte estendido do ElastiCache.

Responsabilidades do Amazon ElastiCache

Após a data de término do suporte padrão do ElastiCache, o Amazon ElastiCache fornecerá patches, correções de erros e atualizações para mecanismos inscritos no Suporte estendido do ElastiCache. Isso ocorrerá por até três anos ou até que você pare de usar os mecanismos no Suporte estendido, o que ocorrer primeiro.

Suas responsabilidades

Você é responsável por aplicar os patches, correções de erros e atualizações fornecidos para caches no Suporte estendido do ElastiCache. O Amazon ElastiCache reserva-se o direito de alterar, substituir ou retirar esses patches, correções de erros e atualizações a qualquer momento. Caso seja necessário um patch para resolver problemas críticos de segurança ou estabilidade, o Amazon ElastiCache reserva-se o direito de atualizar seus caches com o patch ou exigir que você instale o patch.

Você também é responsável por atualizar o mecanismo para uma versão mais nova antes da data de término do Suporte estendido do ElastiCache. A data de término do Suporte estendido do ElastiCache normalmente é três anos após a data de término do suporte padrão do ElastiCache.

Se você não atualizar o mecanismo, após o fim do Suporte estendido do ElastiCache, o Amazon ElastiCache tentará atualizá-lo para a versão mais recente aceita pelo suporte padrão do ElastiCache. Se a atualização falhar, o Amazon ElastiCache se reserva o direito de excluir o cache que está executando o mecanismo após a data de término do suporte padrão do ElastiCache. No entanto, antes de fazer isso, o Amazon ElastiCache preservará os dados desse mecanismo.

Gerenciamento de versões para ElastiCache

Gerencie como você gostaria de atualizar seus ElastiCache caches e clusters baseados em nós atualizados para os mecanismos OSS Valkey, Memcached e Redis.

Gerenciamento de versões para ElastiCache cache sem servidor

Gerencie se e quando o cache ElastiCache sem servidor é atualizado e realize atualizações de versão de acordo com seus próprios termos e cronogramas.

ElastiCache O Serverless aplica automaticamente a versão mais recente do software secundário e de patch ao seu cache, sem nenhum impacto ou tempo de inatividade em seu aplicativo. Não é necessária nenhuma ação de sua parte.

Quando uma nova versão principal estiver disponível, o ElastiCache Serverless enviará uma notificação no console e um evento no EventBridge. Você pode optar por atualizar o cache para a versão principal mais recente modificando o cache usando o console, a CLI ou a API e selecionando a versão mais recente do mecanismo. Semelhante às atualizações secundárias e de patch, as atualizações da versão principal são realizadas sem tempo de inatividade do seu aplicativo.

Gerenciamento de versões para clusters baseados em nós ElastiCache

Ao trabalhar com ElastiCache clusters baseados em nós, você pode controlar quando o software que alimenta seu cluster é atualizado para novas versões suportadas pelo ElastiCache. É possível controlar quando atualizar o cache para as versões MAJOR, MINOR e PATCH mais recentes disponíveis. Você inicia atualizações de versão do mecanismo no seu cluster ou grupo de replicação, modificando-o e especificando uma nova versão do mecanismo.

Você pode controlar se e quando o software compatível com o protocolo que alimenta seu cluster é atualizado para novas versões suportadas pelo ElastiCache. Esse nível de controle permite que você mantenha a compatibilidade com versões específicas, teste novas versões com seu aplicativo antes de implantar em produção e realize atualizações de versão em seus próprios termos e cronogramas.

Como as atualizações de versões podem envolver algum risco de compatibilidade, elas não ocorrem automaticamente. Você deve iniciá-las.

Clusters Valkey e Redis OSS

Note

- Se um cluster Valkey ou Redis OSS for replicado em uma ou mais regiões, a versão do mecanismo será atualizada para regiões secundárias e depois para a região primária.
- ElastiCache para Redis, as versões OSS são identificadas com uma versão semântica que compreende um componente maior e um menor. Por exemplo, no Redis OSS 6.2, a versão principal é a 6, a versão secundária é a 2. Ao operar clusters baseados em nós, ElastiCache para Redis OSS também expõe o componente de patch, por exemplo, Redis OSS 6.2.1, e a versão do patch é 1.

As versões principais são para alterações incompatíveis da API e as versões secundárias são para novas funcionalidades adicionadas de forma compatível com as versões anteriores. As versões do patch são para correções de bugs compatíveis com versões anteriores e mudanças não funcionais.

Com o Valkey e Redis OSS, você inicia atualizações de versão do mecanismo no seu cluster ou grupo de replicação, modificando-o e especificando uma nova versão do mecanismo. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

Memcached

Com o Memcached, para atualizar para uma versão mais recente, você deve modificar seu cluster e especificar a nova versão do mecanismo que deseja usar. Atualizar para uma versão do Memcached mais recente é um processo destrutivo. Você perde seus dados e começa com um cache frio. Para obter mais informações, consulte [Modificação de um cluster do ElastiCache](#).

Você deve estar ciente dos seguintes requisitos ao atualizar de uma versão mais antiga do Memcached para o Memcached versão 1.4.33 ou posterior. `CreateCacheCluster` e `ModifyCacheCluster` falham nas seguintes condições:

- Se `slab_chunk_max > max_item_size`.
- Se `max_item_size` modulo `slab_chunk_max` $\neq 0$.
- Se `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

O valor `(max_cache_memory - memcached_connections_overhead)` é a memória do nó utilizável para dados. Para obter mais informações, consulte [Sobrecarga de conexões do Memcached](#).

Mecanismos e versões compatíveis

ElastiCache caches sem servidor oferecem suporte à ElastiCache versão 7.2 para Valkey e superior, ElastiCache versão 1.6 para Memcached e superior e 7.0 para Redis OSS e ElastiCache superior.

ElastiCache Os clusters baseados em nós oferecem suporte à ElastiCache versão 7.2 para Valkey e superior, ElastiCache versão 1.4.5 para Memcached e superior e 4.0.10 para Redis OSS e ElastiCache superior.

Os ElastiCache clusters baseados em nós oferecem suporte às seguintes versões do Valkey:

- [Versões compatíveis do Valkey](#)
- [ElastiCache versão 8.2 para Valkey](#)
- [ElastiCache versão 8.1 para Valkey](#)
- [ElastiCache versão 8.0 para Valkey](#)
- [ElastiCache versão 7.2.6 para Valkey](#)

Versões compatíveis do Valkey

Versões compatíveis do Valkey abaixo. Observe que o Valkey oferece suporte à maioria dos recursos disponíveis na ElastiCache versão 7.2 para Redis OSS por padrão.

- Você também pode atualizar seus ElastiCache clusters com versões anteriores à 5.0.6. O processo envolvido é o mesmo, mas pode incorrer em tempo de failover mais longo durante a propagação do DNS (30 s a 1 m).
- A partir do Redis OSS 7, ElastiCache oferece suporte à alternância entre Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado).
- O processo de atualização do mecanismo Amazon ElastiCache for Redis OSS foi projetado para fazer o melhor esforço para reter seus dados existentes e requer uma replicação bem-sucedida do Redis OSS.
- Ao atualizar o mecanismo, ElastiCache encerrará as conexões existentes do cliente. Para minimizar o tempo de inatividade durante as atualizações do mecanismo, recomendamos implementar [as práticas recomendadas para clientes do Redis OSS](#) com novas tentativas de erro e recuo exponencial e as práticas recomendadas para [minimizar o tempo de inatividade durante a manutenção](#).
- Não é possível atualizar diretamente do Valkey ou Redis OSS (modo cluster desabilitado) para o Valkey ou Redis OSS (modo cluster habilitado) ao atualizar seu mecanismo. O procedimento a seguir mostra como atualizar do Valkey ou Redis OSS (modo cluster desabilitado) para o Valkey ou Redis OSS (modo cluster habilitado).

Para atualizar de uma versão de mecanismo Valkey ou Redis OSS (modo cluster desabilitado) para Valkey ou Redis OSS (modo cluster habilitado)

1. Faça um backup do seu cluster do Valkey ou Redis OSS (modo cluster desabilitado) ou do grupo de replicação. Para obter mais informações, consulte [Realização de backups manuais](#).
2. Use o backup para criar e propagar um cluster do Valkey ou Redis OSS (modo cluster habilitado) com um fragmento (grupo de nós). Especifique a nova versão do mecanismo e habilite o modo de cluster ao criar o cluster ou o grupo de replicação. Para obter mais informações, consulte [Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente](#).
3. Exclua o antigo cluster do Valkey ou Redis OSS (modo cluster desabilitado) ou o grupo de replicação. Para acessar mais informações, consulte [Excluindo um cluster no ElastiCache](#) ou [Exclusão de um grupo de replicação](#).

4. Escale o novo cluster ou grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) para o número de fragmentos (grupos de nós) que você precisa. Para obter mais informações, consulte [Escalaabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).
- Ao atualizar as principais versões do mecanismo, por exemplo, de 5.0.6 para 6.0, também é necessário selecionar um novo grupo de parâmetros que seja compatível com a nova versão do mecanismo.
- Para clusters Redis OSS de nó único e clusters Redis individuais com o Multi-AZ desabilitado, recomendamos que seja disponibilizada memória suficiente para o Redis OSS, conforme descrito em [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#). Nesses casos, o primário não está disponível para solicitações de serviço durante o processo de atualização.
- Para clusters Redis OSS com o Multi-AZ habilitado, também recomendamos que você programe atualizações do mecanismo durante períodos de baixo tráfego de gravações recebidas. Ao atualizar para o Redis OSS 5.0.6 ou posterior, o cluster primário continua disponível para solicitações de serviço durante o processo de atualização.

Os clusters e os grupos de replicação com vários fragmentos são processados e corrigidos da seguinte forma:

- Todos os estilhaços são processados em paralelo. Somente uma operação de atualização é realizada em um estilhaço por vez.
- Em cada fragmento, todas as réplicas são processadas antes do processamento da primária. Caso haja menos réplicas em um fragmento, a primária nesse fragmento pode ser processada antes da conclusão do processamento das réplicas em outros fragmentos.
- Em todos os fragmentos, os nós primários são processados em série. Somente um nó primário é atualizado por vez.
- Caso a criptografia esteja habilitada no cluster ou no grupo de replicação atual, não será possível atualizar para uma versão de mecanismo que não ofereça suporte à criptografia, como de 3.2.6 a 3.2.10.

Considerações sobre o Memcached

Ao atualizar um cluster Memcached baseado em nós, considere o seguinte.

- O gerenciamento da versão do mecanismo foi desenvolvido para que você possa ter o máximo controle possível sobre a execução de patches. No entanto, ElastiCache se reserva o direito de

corrigir seu cluster em seu nome no caso improvável de uma vulnerabilidade crítica de segurança no sistema ou no software de cache.

- Como o mecanismo Memcached não oferece suporte para persistência, as atualizações de versão do mecanismo Memcached são sempre um processo disruptivo que limpa todos os dados do cache no cluster.

ElastiCache versão 8.2 para Valkey

Aqui estão alguns dos novos recursos introduzidos no Valkey 8.2 (em comparação com o ElastiCache Valkey 8.1):

- Compatibilidade nativa para [pesquisa vetorial](#), permitindo o armazenamento, a indexação, pesquisa e atualização de bilhões de incorporações vetoriais de alta dimensão na memória com latências de até microssegundos.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para saber mais sobre a versão 8.2 do Valkey, introduzindo a pesquisa vetorial, consulte [Valkey Search](#).

ElastiCache versão 8.1 para Valkey

Aqui estão alguns dos novos recursos introduzidos no Valkey 8.1 (em comparação com o ElastiCache Valkey 8.0):

- Uma [nova implementação de tabela de hash](#) que reduz a sobrecarga de memória para reduzir o uso de memória em até 20% para padrões comuns key/value .
- Compatibilidade nativa para [filtros Bloom](#), um novo tipo de dados que permite realizar pesquisas usando até 98% menos memória em comparação com o uso do tipo de dados Set.
- Novo comando [COMMANDLOG](#) que registra execuções lentas, solicitações grandes e respostas grandes.
- Novo suporte de atualização condicional para o comando SET usando o argumento IFEQ.
- Melhorias de desempenho, incluindo latência até 45% menor para o comando ZRANK, desempenho até 12x mais rápido para PFMERGE e PFCOUNT e throughput até 514% maior para BITCOUNT.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para saber mais sobre a versão 8.1 do Valkey, consulte [Notas de versão do Valkey 8.1](#)

ElastiCache versão 8.0 para Valkey

Aqui estão alguns dos novos recursos introduzidos no Valkey 8.0 (em comparação com o ElastiCache Valkey 7.2.6):

- Melhorias na eficiência da memória, permitindo que os usuários armazenem até 20% mais dados por nó sem nenhuma alteração no aplicativo.
- Infraestrutura de métricas por slot recém-introduzida para clusters baseados em nós, fornecendo visibilidade detalhada do desempenho e do uso de recursos de slots individuais.
- ElastiCache O Serverless for Valkey 8.0 pode dobrar as solicitações suportadas por segundo (RPS) a cada 2-3 minutos, atingindo 5 milhões de RPS por cache de zero em menos de 13 minutos, com latência de leitura p50 consistente abaixo de um milissegundo.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para obter mais informações sobre a versão 8 do Valkey, consulte [Notas de versão do Valkey 8](#)

ElastiCache versão 7.2.6 para Valkey

Em 10 de outubro de 2024, a ElastiCache versão 7.2.6 do Valkey foi lançada. Aqui estão alguns dos novos recursos introduzidos na versão 7.2 (em comparação com a ElastiCache versão 7.1 do Redis OSS):

- Otimizações de desempenho e memória para vários tipos de dados: otimização de memória para chaves de tipo de lista e conjunto, otimização de velocidade para comandos de conjuntos ordenados, otimização de desempenho para comandos com várias teclas no modo de cluster, melhorias de pub/sub desempenho, otimização de desempenho para comandos SCAN, SSCAN, HSCAN, ZSCAN e várias outras otimizações menores.
- Nova opção WITHSCORE para os comandos ZRANK e ZREVRANK
- CLIENT NO-TOUCH para que os clientes executem comandos sem afetar LRU/LFU as teclas.
- Novo comando CLUSTER MYSHARDID que retorna o ID do fragmento do nó para agrupar logicamente os nós no modo de cluster com base na replicação.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para obter mais informações sobre a ElastiCache versão 7.2 para a versão Valkey, consulte as [notas de lançamento do Redis OSS 7.2.4](#) (a versão 7.2 para Valkey inclui todas as alterações da ElastiCache versão 7.1 para Redis OSS até a ElastiCache versão 7.2.4 para Redis OSS). ElastiCache [Notas de lançamento do Valkey 7.2](#) no Valkey on. GitHub

ElastiCache versão 8.2 para Valkey

Aqui estão alguns dos novos recursos introduzidos no Valkey 8.2 (em comparação com o ElastiCache Valkey 8.1):

- Compatibilidade nativa para [pesquisa vetorial](#), permitindo o armazenamento, a indexação, pesquisa e atualização de bilhões de incorporações vetoriais de alta dimensão na memória com latências de até microssegundos.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para saber mais sobre a versão 8.2 do Valkey, introduzindo a pesquisa vetorial, consulte [Valkey Search](#).

ElastiCache versão 8.1 para Valkey

Aqui estão alguns dos novos recursos introduzidos no Valkey 8.1 (em comparação com o ElastiCache Valkey 8.0):

- Uma [nova implementação de tabela de hash](#) que reduz a sobrecarga de memória para reduzir o uso de memória em até 20% para padrões comuns key/value .
- Compatibilidade nativa para [filtros Bloom](#), um novo tipo de dados que permite realizar pesquisas usando até 98% menos memória em comparação com o uso do tipo de dados Set.
- Novo comando [COMMANDLOG](#) que registra execuções lentas, solicitações grandes e respostas grandes.
- Novo suporte de atualização condicional para o comando SET usando o argumento IFEQ.
- Melhorias de desempenho, incluindo latência até 45% menor para o comando ZRANK, desempenho até 12x mais rápido para PFMERGE e PFCOUNT e throughput até 514% maior para BITCOUNT.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para saber mais sobre a versão 8.1 do Valkey, consulte [Notas de versão do Valkey 8.1](#)

ElastiCache versão 8.0 para Valkey

Aqui estão alguns dos novos recursos introduzidos no Valkey 8.0 (em comparação com o ElastiCache Valkey 7.2.6):

- Melhorias na eficiência da memória, permitindo que os usuários armazenem até 20% mais dados por nó sem nenhuma alteração no aplicativo.
- Infraestrutura de métricas por slot recém-introduzida para clusters baseados em nós, fornecendo visibilidade detalhada do desempenho e do uso de recursos de slots individuais.
- ElastiCache O Serverless for Valkey 8.0 pode dobrar as solicitações suportadas por segundo (RPS) a cada 2-3 minutos, atingindo 5 milhões de RPS por cache de zero em menos de 13 minutos, com latência de leitura p50 consistente abaixo de um milissegundo.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para obter mais informações sobre a versão 8 do Valkey, consulte [Notas de versão do Valkey 8](#)

ElastiCache versão 7.2.6 para Valkey

Em 10 de outubro de 2024, a ElastiCache versão 7.2.6 do Valkey foi lançada. Aqui estão alguns dos novos recursos introduzidos na versão 7.2 (em comparação com a ElastiCache versão 7.1 do Redis OSS):

- Otimizações de desempenho e memória para vários tipos de dados: otimização de memória para chaves de tipo de lista e conjunto, otimização de velocidade para comandos de conjuntos ordenados, otimização de desempenho para comandos com várias teclas no modo de cluster, melhorias de pub/sub desempenho, otimização de desempenho para comandos SCAN, SSCAN, HSCAN, ZSCAN e várias outras otimizações menores.
- Nova opção WITHSCORE para os comandos ZRANK e ZREVRANK
- CLIENT NO-TOUCH para que os clientes executem comandos sem afetar LRU/LFU as teclas.
- Novo comando CLUSTER MYSHARDID que retorna o ID do fragmento do nó para agrupar logicamente os nós no modo de cluster com base na replicação.

Para ter mais informações sobre o Valkey, consulte [Valkey](#).

Para obter mais informações sobre a ElastiCache versão 7.2 para a versão Valkey, consulte as [notas de lançamento do Redis OSS 7.2.4](#) (a versão 7.2 para Valkey inclui todas as alterações da ElastiCache versão 7.1 para Redis OSS até a ElastiCache versão 7.2.4 para Redis OSS). ElastiCache [Notas de lançamento do Valkey 7.2](#) no Valkey on. GitHub

Versões compatíveis do mecanismo do Redis OSS

ElastiCache Caches sem servidor e clusters baseados em nós oferecem suporte a todas as versões 7.1 e anteriores do Redis OSS.

- [ElastiCache versão 7.1 para Redis OSS \(aprimorada\)](#)

Os ElastiCache clusters baseados em nós oferecem suporte às seguintes versões do Redis OSS:

- [ElastiCache versão 7.1 para Redis OSS \(aprimorada\)](#)
- [ElastiCache versão 7.0 para Redis OSS \(aprimorada\)](#)
- [ElastiCache versão 6.2 para Redis OSS \(aprimorada\)](#)
- [ElastiCache versão 6.0 para Redis OSS \(aprimorada\)](#)
- [ElastiCache versão 5.0.6 para Redis OSS \(aprimorada\)](#)
- [ElastiCache versão 5.0.5 para Redis OSS \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache versão 5.0.4 para Redis OSS \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache versão 5.0.3 para Redis OSS \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache versão 5.0.0 para Redis OSS \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache versão 4.0.10 para Redis OSS \(aprimorada\)](#)
- [Versões que ultrapassaram o fim de vida útil \(EOL\) \(3.x\)](#)
- [Versões que ultrapassaram o fim de vida útil \(EOL\) \(2.x\)](#)

ElastiCache versão 7.1 para Redis OSS (aprimorada)

Esta versão contém melhorias de desempenho que permitem que as cargas de trabalho aumentem a produtividade e reduzam as latências de operação. ElastiCache [a versão 7.1 para Redis OSS apresenta dois aprimoramentos principais](#):

Estendemos a funcionalidade aprimorada de I/O threads para também lidar com a lógica da camada de apresentação. Por camada de apresentação, queremos dizer que os threads de E/S aprimorados

agora não estão apenas lendo a entrada do cliente, mas também analisando a entrada no formato de comando binário do Redis OSS. Em seguida, isso é encaminhado para o thread principal para execução, o que proporciona um ganho de desempenho. Padrão de acesso à memória do Redis OSS aprimorado. As etapas de execução de muitas operações da estrutura de dados são intercaladas, para garantir acesso paralelo à memória e reduzir a latência do acesso à memória. Ao executar ElastiCache em Graviton3 R7g.4xlarge ou maior, os clientes podem atingir mais de 1 milhão de solicitações por segundo por nó. Com as melhorias de desempenho do Redis OSS v7.1, os clientes podem obter até 100% mais taxa de transferência e 50% menos latência P99 em relação ao Redis OSS v7.0. ElastiCache ElastiCache Esses aprimoramentos são habilitados em tamanhos de nós com pelo menos oito núcleos físicos (2xlarge no Graviton e 4xlarge no x86), independentemente do tipo de CPU, e não exigem alterações no cliente.

 Note

ElastiCache A v7.1 é compatível com o Redis OSS v7.0.

ElastiCache versão 7.0 para Redis OSS (aprimorada)

ElastiCache para Redis, o OSS 7.0 adiciona uma série de melhorias e suporte para novas funcionalidades:

- [Funções](#): ElastiCache para Redis, o OSS 7 adiciona suporte às funções do Redis OSS e fornece uma experiência gerenciada que permite que os desenvolvedores executem [scripts LUA](#) com a lógica do aplicativo armazenada no ElastiCache cluster, sem exigir que os clientes reenviem os scripts para o servidor a cada conexão.
- [Melhorias na ACL](#): o Valkey e o Redis OSS 7 adicionam suporte para a próxima versão das listas de controle de acesso (). ACLs Os clientes agora podem especificar vários conjuntos de permissões em chaves ou espaços de chave específicos no Valkey e Redis OSS.
- [Sharded Pub/Sub](#): ElastiCache para Valkey e Redis, o OSS 7 adiciona suporte para executar Pub/Sub functionality in a sharded way when running ElastiCache in Cluster Mode Enabled (CME). Pub/Sub recursos que permitem que os editores enviem mensagens para qualquer número de inscritos em um canal. Os canais são vinculados a um fragmento no ElastiCache cluster, eliminando a necessidade de propagar as informações do canal entre os fragmentos, resultando em maior escalabilidade.
- I/O Multiplexação aprimorada: ElastiCache para Valkey e Redis, o OSS 7 introduz a multiplexação de E/S aprimorada, que oferece maior taxa de transferência e latência reduzida para cargas

de trabalho de alto rendimento que têm muitas conexões simultâneas de clientes com um cluster. ElastiCache Por exemplo, ao usar um cluster de nós r6g.xlarge e executar 5200 clientes simultâneos, você pode obter um aumento de até 72% na taxa de transferência (operações de leitura e gravação por segundo) e uma redução de até 71% na latência P99, em comparação com a versão 6 do Redis OSS. ElastiCache

Para ter mais informações sobre o Valkey, consulte [Valkey](#). Para obter mais informações sobre a versão 7.0 do Redis OSS, consulte as notas de lançamento do [Redis OSS 7.0 em Redis OSS](#) on. GitHub

ElastiCache versão 6.2 para Redis OSS (aprimorada)

ElastiCache para Redis, o OSS 6.2 inclui melhorias de desempenho para clusters habilitados para TLS usando tipos de nós x86 com 8 v CPUs ou mais ou tipos de nós Graviton2 com 4 v ou mais. CPUs Esses aprimoramentos melhoram a taxa de transferência e reduzem o tempo de estabelecimento da conexão do cliente, transferindo a criptografia para outros v. CPUs Com o Redis OSS 6.2, você também pode gerenciar o acesso aos canais do Pub/Sub com as regras da Lista de Controle de Acesso (ACL).

Com essa versão, também introduzimos suporte para armazenamento de dados em camadas em nós de cluster contendo NVMe SSD conectado localmente. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

A versão 6.2.6 do mecanismo Redis OSS também introduz suporte ao formato nativo de notação de JavaScript objeto (JSON), uma maneira simples e sem esquemas de codificar conjuntos de dados complexos dentro dos clusters do Redis OSS. Com o suporte a JSON, você pode aproveitar o desempenho e o Redis OSS APIs para aplicativos que operam em JSON. Para obter mais informações, consulte [Conceitos básicos do JSON](#). Também estão incluídas métricas relacionadas ao JSON `JsonBasedCmds` e `JsonBasedCmdsLatency`, que são incorporadas CloudWatch para monitorar o uso desse tipo de dados. Para obter mais informações, consulte [Métricas para o Valkey e Redis OSS](#).

Você especifica a versão do mecanismo usando 6.2. ElastiCache invocará automaticamente a versão de patch preferida do Redis OSS 6.2 que está disponível. Por exemplo, ao criar/modificar um cluster, você define o parâmetro `--engine-version` como 6.2. O cluster será lançado com a versão atual de patch preferencial disponível do Redis OSS 6.2 no momento. `creation/modification` Especificar a versão 6.x do mecanismo na API resultará na versão secundária mais recente do Redis OSS 6.

Para clusters 6.0 existentes, você pode optar pela próxima atualização automática da versão secundária definindo o `AutoMinorVersionUpgrade` parâmetro como `yes` em `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` ou `ModifyReplicationGroup` APIs ElastiCache atualizará a versão secundária de seus clusters 6.0 existentes para 6.2 usando atualizações de autoatendimento. Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache](#).

Ao chamar a `DescribeCacheEngineVersions` API, o valor do `EngineVersion` parâmetro será definido como 6.2 e a versão real do mecanismo com a versão do patch será retornada no `CacheEngineVersionDescription` campo.

Para obter mais informações sobre a versão 6.2 do Redis OSS, consulte as notas de lançamento do [Redis OSS 6.2 em Redis OSS](#) on. GitHub

ElastiCache versão 6.0 para Redis OSS (aprimorada)

A Amazon ElastiCache apresenta a próxima versão do ElastiCache mecanismo Redis OSS, que inclui [autenticação de usuários com controle de acesso baseado em funções](#), armazenamento em cache do lado do cliente e melhorias operacionais significativas.

A partir do Redis OSS 6.0, ElastiCache oferecerá uma única versão para cada versão secundária do Redis OSS, em vez de oferecer várias versões de patch. ElastiCache gerenciará automaticamente a versão de patch de seus clusters em execução, garantindo melhor desempenho e segurança aprimorada.

Você também pode optar pela próxima atualização automática da versão secundária definindo o `AutoMinorVersionUpgrade` parâmetro como `yes` e ElastiCache gerenciará a atualização da versão secundária, por meio de atualizações de autoatendimento. Para obter mais informações, consulte [Atualizações de serviço em ElastiCache](#).

Você especifica a versão do mecanismo usando `6.0`. ElastiCache invocará automaticamente a versão de patch preferida do Redis OSS 6.0 que está disponível. Por exemplo, quando você é `create/modify` um cluster, você define o `--engine-version` parâmetro como 6.0. O cluster será iniciado com a versão de patch preferencial atual disponível do Redis OSS 6.0 no momento da criação/modificação. Qualquer solicitação com um valor específico de versão de patch será rejeitada, uma exceção será lançada e o processo falhará.

Ao chamar a `DescribeCacheEngineVersions` API, o valor do `EngineVersion` parâmetro será definido como 6.0 e a versão real do mecanismo com a versão do patch será retornada no `CacheEngineVersionDescription` campo.

Para obter mais informações sobre a versão 6.0 do Redis OSS, consulte as notas de lançamento do [Redis OSS 6.0 em Redis OSS](#) on. GitHub

ElastiCache versão 5.0.6 para Redis OSS (aprimorada)

A Amazon ElastiCache apresenta a próxima versão do ElastiCache mecanismo Redis OSS, que inclui correções de bugs e as seguintes atualizações cumulativas:

- Garantia de estabilidade do mecanismo em condições especiais.
- Tratamento aprimorado de erros do Hyperloglog.
- Comandos de handshake aprimorados para a replicação confiável.
- Rastreamento de entrega de mensagens consistente pelo comando XCLAIM.
- Gerenciamento do campo LFU aprimorado nos objetos.
- Gerenciamento de transações aprimorado ao usar ZPOP.
- Capacidade de renomear comandos: um parâmetro denominado `rename-commands` que permite renomear comandos do Redis OSS possivelmente perigosos e caros que podem causar a perda acidental de dados, como FLUSHALL ou FLUSHDB. Isso é semelhante à configuração `rename-command` no Redis OSS de software livre. No entanto, ElastiCache melhorou a experiência ao fornecer um fluxo de trabalho totalmente gerenciado. As alterações de nomes de comando são aplicadas imediatamente e propagadas automaticamente em todos os nós no cluster que contêm a lista de comandos. Não há nenhuma intervenção necessária de sua parte, como a reinicialização de nós.

Os exemplos a seguir demonstram como modificar os grupos de parâmetros existentes. Eles incluem o parâmetro `rename-commands`, que é uma lista separada por espaços de comandos que você deseja renomear:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

Neste exemplo, o parâmetro `rename-commands` é usado para renomear o comando `flushall` como `restrictedflushall`.

Para renomear vários comandos, use o seguinte:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-  
name custom_param_group  
--parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall  
restrictedflushall flushdb restrictedflushdb'" --region region
```

Para reverter qualquer alteração, execute novamente o comando e exclua qualquer valor renomeado na lista `ParameterValue` que você deseja manter, conforme mostrado a seguir:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-  
name custom_param_group  
--parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall  
restrictedflushall'" --region region
```

Nesse caso, o comando `flushall` é renomeado para `restrictedflushall` e qualquer outro comando renomeado revertido para seus nomes de comando originais.

Note

Ao renomear comandos, você está restrito às seguintes limitações:

- Todos os comandos renomeados devem ser alfanuméricos.
- O tamanho máximo de novos nomes de comando é de 20 caracteres alfanuméricos.
- Ao renomear comandos, certifique-se de que você atualize o grupo de parâmetros associado ao seu cluster.
- Para evitar totalmente o uso de um comando, use a palavra-chave `blocked`, conforme mostrado a seguir:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-  
name custom_param_group  
--parameter-name-values "ParameterName=rename-commands,  
ParameterValue='flushall blocked'" --region region
```

Para obter mais informações sobre as alterações de parâmetro e uma lista de comandos que são qualificados para serem renomeados, consulte [Alterações de parâmetros do Redis OSS 5.0.3](#).

- Redis OSS Streams: essa opção modela uma estrutura de dados de log que permite que os produtores anexem novos itens em tempo real. Também permite que os consumidores leiam

mensagens com ou sem bloqueio. Os fluxos também permitem grupos de consumidores, que representam um grupo de clientes para cooperativamente consumir diferentes partes do mesmo fluxo de mensagens, semelhante ao [Apache Kafka](#). Para obter mais informações, consulte [Fluxos](#).

- Oferece suporte para uma família de comandos de fluxo, como os XADD, XRANGE e XREAD. Para obter mais informações, consulte [Comandos dos fluxos](#).
- Diversos parâmetros novos e renomeados. Para obter mais informações, consulte [Alterações de parâmetros do Redis OSS 5.0.0](#).
- Uma nova métrica do Redis OSS, `StreamBasedCmds`.
- Tempo de snapshot um pouco mais rápido para nós Redis OSS.

Important

ElastiCache retrocedeu duas correções de bugs críticos da versão 5.0.1 de [código aberto do Redis OSS](#). Eles estão listados a seguir:

- RESTAURAR resposta incompatível quando determinadas chaves já expiraram.
- O comando XCLAIM pode retornar uma entrada errada ou dessincronizar o protocolo.

Ambas as correções de bugs estão incluídas no suporte do Redis OSS ElastiCache para a versão 5.0.0 do mecanismo Redis OSS e são consumidas em futuras atualizações da versão.

Para obter mais informações, consulte as [notas de lançamento do Redis OSS 5.0.6](#) em Redis OSS on. GitHub

ElastiCache versão 5.0.5 para Redis OSS (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache apresenta a próxima versão do motor ElastiCache Redis OSS;. Ele inclui alterações de configuração on-line para clusters ElastiCache de failover automático durante todas as operações planejadas. Agora é possível escalar seu cluster, atualizar a versão do mecanismo Redis OSS e aplicar patches e atualizações de manutenção enquanto o cluster permanece online e continua atendendo às solicitações recebidas. Ela também inclui correções de erros.

Para obter mais informações, consulte as [notas de lançamento do Redis OSS 5.0.5](#) em Redis OSS on. GitHub

ElastiCache versão 5.0.4 para Redis OSS (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache apresenta a próxima versão do mecanismo Redis OSS compatível com. ElastiCache Ele inclui as melhorias a seguir:

- Garantia de estabilidade do mecanismo em condições especiais.
- Tratamento aprimorado de erros do Hyperloglog.
- Comandos de handshake aprimorados para a replicação confiável.
- Rastreamento de entrega de mensagens consistente pelo comando XCLAIM.
- Gerenciamento do campo LFU aprimorado nos objetos.
- Gerenciamento de transações aprimorado ao usar ZPOP.

Para obter mais informações, consulte as [notas de lançamento do Redis OSS 5.0.4](#) em Redis OSS on. GitHub

ElastiCache versão 5.0.3 para Redis OSS (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache apresenta a próxima versão do ElastiCache mecanismo Redis OSS, que inclui correções de bugs.

ElastiCache versão 5.0.0 para Redis OSS (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache apresenta a próxima versão principal do mecanismo ElastiCache Redis OSS. ElastiCache A versão 5.0.0 para Redis OSS oferece suporte para as seguintes melhorias:

- Redis OSS Streams: essa opção modela uma estrutura de dados de log que permite que os produtores anexem novos itens em tempo real. Também permite que os consumidores leiam mensagens com ou sem bloqueio. Os fluxos também permitem grupos de consumidores, que representam um grupo de clientes para cooperativamente consumir diferentes partes do mesmo fluxo de mensagens, semelhante ao [Apache Kafka](#). Para obter mais informações, consulte [Fluxos](#).
- Oferece suporte para uma família de comandos de fluxo, como os XADD, XRANGE e XREAD. Para obter mais informações, consulte [Comandos dos fluxos](#).
- Diversos parâmetros novos e renomeados. Para obter mais informações, consulte [Alterações de parâmetros do Redis OSS 5.0.0](#).
- Uma nova métrica do Redis OSS, `StreamBasedCmds`.
- Tempo de snapshot um pouco mais rápido para nós Redis OSS.

ElastiCache versão 4.0.10 para Redis OSS (aprimorada)

A Amazon ElastiCache apresenta a próxima versão principal do mecanismo ElastiCache Redis OSS. ElastiCache a versão 4.0.10 para Redis OSS traz suporte para as seguintes melhorias:

- Redimensionamento e criptografia de clusters on-line em uma única ElastiCache versão. Para saber mais, consulte:
 - [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#)
 - [Refragmentação online para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#)
 - [Segurança de dados no Amazon ElastiCache](#)
- Diversos novos parâmetros. Para obter mais informações, consulte [Alterações de parâmetros do Redis OSS 4.0.10](#).
- Suporte para a família de comandos de memória, como MEMORY. Para obter mais informações, consulte [Comandos](#) (pesquisar em MEMÓRIA).
- Suporte para desfragmentação de memória online, permitindo assim a utilização mais eficiente da memória e uma quantidade maior de memória disponível para seus dados.
- Support para descargas e exclusões assíncronas. ElastiCache para Redis, o OSS suporta comandos como UNLINK, FLUSHDB e FLUSHALL para execução em um encadeamento diferente do encadeamento principal. Fazer isso ajuda a melhorar o desempenho e os tempos de resposta dos aplicativos, liberando a memória de maneira assíncrona.
- Uma nova métrica do Redis OSS, ActiveDefragHits. Para obter mais informações, consulte [Métricas do Redis OSS](#).

Usuários do Redis OSS (modo de cluster desativado) que executam a ElastiCache versão 3.2.10 para Redis OSS podem usar o console para atualizar seus clusters por meio de upgrade online.

Comparando o suporte ao redimensionamento e criptografia de ElastiCache clusters

Recurso	3.2.6	3.2.10	4.0.10 e posterior
Redimensionamento de cluster online *	Não	Sim	Sim
Criptografia em trânsito **	Sim	Não	Sim
Criptografia em repouso **	Sim	Não	Sim

Recurso	3.2.6	3.2.10	4.0.10 e posterior

* Adição, remoção e rebalanceamento de fragmentos.

** Obrigatório para aplicativos compatíveis com FedRAMP, HIPAA e PCI DSS. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

Versões que ultrapassaram o fim de vida útil (EOL) (3.x)

ElastiCache versão 3.2.10 para Redis OSS (aprimorada)

A Amazon ElastiCache apresenta a próxima versão principal do mecanismo ElastiCache Redis OSS. ElastiCache a versão 3.2.10 para Redis OSS (aprimorada) introduz o redimensionamento on-line do cluster para adicionar ou remover fragmentos do cluster enquanto ele continua atendendo às solicitações recebidas. I/O ElastiCache para Redis OSS 3.2.10, os usuários têm todas as funcionalidades das versões anteriores do Redis OSS, exceto a capacidade de criptografar seus dados. Esse recurso está atualmente disponível apenas na versão 3.2.6.

Comparando ElastiCache as versões 3.2.6 e 3.2.10 para Redis OSS

Recurso	3.2.6	3.2.10
Redimensionamento de cluster online *	Não	Sim
Criptografia em trânsito **	Sim	Não
Criptografia em repouso **	Sim	Não

* Adição, remoção e rebalanceamento de fragmentos.

** Obrigatório para aplicativos compatíveis com FedRAMP, HIPAA e PCI DSS. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

Para saber mais, consulte:

- [Refragmentação online para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#)
- [Redimensionamento de cluster on-line](#)

ElastiCache versão 3.2.6 para Redis OSS (aprimorada)

A Amazon ElastiCache apresenta a próxima versão principal do mecanismo ElastiCache Redis OSS. ElastiCache A versão 3.2.6 para usuários do Redis OSS tem acesso a todas as funcionalidades das versões anteriores do Redis OSS, além da opção de criptografar seus dados. Para saber mais, consulte:

- [ElastiCache criptografia em trânsito \(TLS\)](#)
- [Criptografia em repouso em ElastiCache](#)
- [Validação de conformidade para a Amazon ElastiCache](#)

ElastiCache versão 3.2.4 para Redis OSS (aprimorada)

A ElastiCache versão 3.2.4 da Amazon apresenta a próxima versão principal do mecanismo ElastiCache Redis OSS. ElastiCache 3.2.4 Os usuários têm todas as funcionalidades das versões anteriores do Redis OSS disponíveis, além da opção de execução no modo cluster ou no modo não cluster. A tabela a seguir fornece um resumo.

Comparação do Redis OSS 3.2.4 modo cluster e modo sem cluster

Recurso	Modo sem cluster	Modo cluster
Particionamento de dados	Não	Sim
Indexação geoespacial	Sim	Sim
Alterar o tipo de nó	Sim	Sim *
Escalabilidade de réplicas	Sim	Sim *
Escalonamento horizontal	Não	Sim *
Suporte para banco de dados	Vários	Único
Grupo de parâmetro	default.redis3.2 **	default.redis3.2.cluster.on **

* Consulte [Restauração de um backup para um novo cache](#)

** Ou uma derivação.

Observações:

- **Particionamento:** a capacidade de dividir seus dados em 2 a 500 grupos de nós (fragmentos) com suporte de replicação para cada grupo de nós.
- **Indexação geoespacial:** o Redis OSS 3.2.4 oferece suporte para a indexação geoespacial por meio de seis comandos GEO. Para obter mais informações, consulte a documentação dos comandos GEO* do Redis OSS [Comandos: GEO](#), na página Valkey Commands (filtrada por GEO).

Para obter informações sobre recursos adicionais do Redis OSS 3, consulte as [notas de versão do Redis OSS 3.2](#) e as [notas de release do Redis OSS 3.0](#).

O Valkey ou Redis OSS atualmente ElastiCache gerenciado (modo de cluster ativado) não oferece suporte aos seguintes recursos do Redis OSS 3.2:

- Migração de réplica
- Rebalanceamento do cluster
- Depurador Lua

ElastiCache desativa os seguintes comandos de gerenciamento do Redis OSS 3.2:

- `cluster meet`
- `cluster replicate`
- `cluster flushslots`
- `cluster addslots`
- `cluster delslots`
- `cluster setslot`
- `cluster saveconfig`
- `cluster forget`
- `cluster failover`
- `cluster bumpepoch`
- `cluster set-config-epoch`
- `cluster reset`

Para obter informações sobre os parâmetros do Redis OSS 3.2.4, consulte [Alterações de parâmetros do Redis OSS 3.2.4](#).

Versões que ultrapassaram o fim de vida útil (EOL) (2.x)

ElastiCache versão 2.8.24 para Redis OSS (aprimorada)

Melhorias do Redis OSS adicionadas desde a versão 2.8.23 incluem correções de bugs e o registro em log de endereços incorretos de acesso à memória. Para obter mais informações, consulte as [Notas de release do Redis OSS 2.8](#).

ElastiCache versão 2.8.23 para Redis OSS (aprimorada)

Melhorias do Redis adicionadas desde a versão 2.8.22 incluem correções de bugs. Para obter mais informações, consulte as [Notas de release do Redis OSS 2.8](#). Esta versão também inclui suporte para o novo parâmetro `close-on-slave-write` que, se ativado, desconecta clientes que tentam gravar em uma réplica somente leitura.

Para obter mais informações sobre os parâmetros do Redis OSS 2.8.23, consulte [Parâmetros adicionados do Redis OSS 2.8.23 \(aprimorado\)](#) o Guia do usuário. ElastiCache

ElastiCache versão 2.8.22 para Redis OSS (aprimorada)

Melhorias do Redis OSS adicionadas desde a versão 2.8.21 incluem o seguinte:

- Suporte para backups e sincronizações sem garantia que permite alocar menos memória para despesas gerais de backup e mais para o seu aplicativo. Para obter mais informações, consulte [Como a sincronização e o backup são implementados](#). O processo sem bifurcação pode afetar tanto a latência quanto o throughput. No caso de alto throughput de gravação, quando uma réplica se sincronizar de novo, ela poderá ficar inacessível durante todo o tempo durante a sincronização.
- Se houver um failover, os grupos de replicação agora se recuperarão mais rapidamente, pois as réplicas realizam sincronizações parciais com o primário em vez de sincronizações completas sempre que possível. Além disso, tanto o primário quanto as réplicas não usam mais o disco durante as sincronizações, proporcionando ganhos de velocidade adicionais.
- Support para duas novas CloudWatch métricas.
 - `ReplicationBytes` - O número de bytes que o cluster primário de um grupo de replicação está enviando às réplicas de leitura.
 - `SaveInProgress` - Um valor binário que indica se há ou não um processo de salvamento em segundo plano em execução.

Para obter mais informações, consulte [Monitorando o uso com CloudWatch métricas](#).

- Uma série de correções críticas de bugs no comportamento PSYNC de replicação. Para obter mais informações, consulte as [Notas de release do Redis OSS 2.8](#).
- Para manter o desempenho aprimorado da replicação em grupos de replicação Multi-AZ e para aumentar a estabilidade do cluster, não há mais suporte para não ElastiCache réplicas.
- Para melhorar a consistência de dados entre o cluster primário e as réplicas em um grupo de replicação, as réplicas não removem mais as chaves independentemente do cluster primário.
- As variáveis de configuração do Redis OSS appendonly e appendfsync não são compatíveis com o Redis OSS versão 2.8.22 e posteriores.
- Em situações de pouca memória, os clientes com um grande buffer de saída podem ser desconectados de um cluster de réplicas. Se desconectado, o cliente deverá se reconectar. Essas situações são mais prováveis para clientes PUBSUB.

ElastiCache versão 2.8.21 para Redis OSS

Melhorias do Redis OSS adicionadas desde a versão 2.8.19 incluem várias correções de bugs. Para obter mais informações, consulte as [Notas de release do Redis OSS 2.8](#).

ElastiCache versão 2.8.19 para Redis OSS

Melhorias do Redis OSS adicionadas desde a versão 2.8.6 incluem o seguinte:

- Support for HyperLogLog. Para obter mais informações, consulte a [nova estrutura de dados do Redis OSS](#):. HyperLogLog
- Agora, o tipo de dados de conjunto classificado oferece suporte para consultas de intervalo lexicográfico com os novos comandos ZRANGEBYLEX, ZLEXCOUNT e ZREMRANGEBYLEX.
- Para evitar que um nó primário envie dados obsoletos para nós de réplica, a SYNC mestre falhará se um processo filho de salvamento (bgsave) for anulado.
- Support para a HyperLogLogBasedCommands CloudWatch métrica. Para obter mais informações, consulte [Métricas para o Valkey e Redis OSS](#).

ElastiCache versão 2.8.6 para Redis OSS

Melhorias do Redis OSS adicionadas desde a versão 2.6.13 incluem o seguinte:

- Resistência aprimorada e tolerância a falhas para réplicas de leitura.

- Suporte para ressincronização parcial.
- Suporte para o número mínimo de réplicas de leitura definido pelo usuário que deve estar disponível em todos os momentos.
- Suporte total para pub/sub-notificação de clientes sobre eventos no servidor.
- Detecção automática de uma falha do nó primário e failover do seu nó primário para um nó secundário.

ElastiCache versão 2.6.13 para Redis OSS

ElastiCache a versão 2.6.13 para Redis OSS foi a versão inicial do Redis OSS compatível ElastiCache . O Multi-AZ não é compatível com a ElastiCache versão 2.6.13 do Redis OSS.

ElastiCache versões para o cronograma de fim de vida útil do Redis OSS

Esta seção define as datas de fim de vida útil (EOL) para as versões principais mais antigas à medida que são anunciadas. Ela ajudará você a tomar decisões sobre versões e atualizações no futuro.

Note

ElastiCache as versões de 5.0.0 a 5.0.5 para Redis OSS estão obsoletas. Use as versões 5.0.6 ou superior.

A tabela a seguir mostra o cronograma do [Extended Support](#) ElastiCache para mecanismos Redis OSS.

Suporte estendido e cronograma de fim da vida útil

Versão principal do mecanismo	Fim do suporte padrão	Início do Suporte estendido premium do ano 1	Início do Suporte estendido premium do ano 2	Início do Suporte estendido premium do ano 3	Fim do Suporte estendido e versão EOL
Redis OSS v4	31/01/2026	01/02/2026	01/02/2027	2/1/2028	31/01/2029

Versão principal do mecanismo	Fim do suporte padrão	Início do Suporte estendido premium do ano 1	Início do Suporte estendido premium do ano 2	Início do Suporte estendido premium do ano 3	Fim do Suporte estendido e versão EOL
Redis OSS v5	31/01/2026	01/02/2026	01/02/2027	2/1/2028	31/01/2029
Redis OSS v6	31/01/2027	01/02/2027	2/1/2028	01/02/2029	31/01/2030

A tabela a seguir resume cada versão e sua data de EOL anunciada, bem como a versão de destino de atualização recomendada.

EOL passado

Versões secundárias de origem	Destino de atualização recomendado	Data de EOL
V3 3.2.4, 3.2.6 e 3.2.10	Versão 6.2 ou superior Note Para as regiões US-ISO-EAST US-ISO-WEST -1, -1 e US-ISOB-EAST -1, recomendamos a versão 5.0.6 ou superior.	31 de julho de 2023
V2 2.8.24, 2.8.23, 2.8.22, 2.8.21, 2.8.19, 2.8.12, 2.8.6, 2.6.13	Versão 6.2 ou superior	13 de janeiro de 2023

Versões secundárias de origem	Destino de atualização recomendado	Data de EOL
	 Note Para as regiões US-ISO-EAST US-ISO-WEST -1, -1 e US-ISOB-EAST -1, recomendamos a versão 5.0.6 ou superior.	

Compatível com ElastiCache versões do Memcached

ElastiCache oferece suporte às seguintes versões do Memcached e à atualização para versões mais recentes. Ao atualizar para uma versão mais recente, fique atento à condições que, caso não sejam atendidas, farão com que a atualização falhe.

ElastiCache para versões do Memcached

- [ElastiCache versão 1.6.22 para Memcached](#)
- [ElastiCache versão 1.6.17 para Memcached](#)
- [ElastiCache versão 1.6.12 para Memcached](#)
- [ElastiCache versão 1.6.6 para Memcached](#)
- [ElastiCache versão 1.5.16 para Memcached](#)
- [ElastiCache versão 1.5.10 para Memcached](#)
- [ElastiCache versão 1.4.34 para Memcached](#)
- [ElastiCache versão 1.4.33 para Memcached](#)
- [ElastiCache versão 1.4.24 para Memcached](#)
- [ElastiCache versão 1.4.14 para Memcached](#)
- [ElastiCache versão 1.4.5 para Memcached](#)

ElastiCache versão 1.6.22 para Memcached

ElastiCache para Memcached versão 1.6.22 para Memcached adiciona suporte para Memcached versão 1.6.22. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.6.18](#).

Para obter mais informações, consulte [ReleaseNotes1622](#) em Memcached on. GitHub

ElastiCache versão 1.6.17 para Memcached

ElastiCache para Memcached versão 1.6.17 for Memcached adiciona suporte para a versão 1.6.17 do mecanismo Memcached. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.6.17](#).

Para obter mais informações, consulte [ReleaseNotes1617](#) em Memcached on. GitHub

ElastiCache versão 1.6.12 para Memcached

ElastiCache para Memcached, a versão 1.6.12 para Memcached adiciona suporte ao mecanismo Memcached 1.6.12 e criptografia em trânsito. Ele inclui correções de erros e atualizações cumulativas do [Memcached 1.6.6](#).

Para obter mais informações, consulte [ReleaseNotes1612](#) em Memcached on. GitHub

ElastiCache versão 1.6.6 para Memcached

ElastiCache para Memcached versão 1.6.6 for Memcached adiciona suporte para Memcached versão 1.6.6. Ele não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.5.16](#). ElastiCache [para Memcached não inclui suporte para Extstore](#).

Para obter mais informações, consulte [ReleaseNotes166](#) em Memcached on. GitHub

ElastiCache versão 1.5.16 para Memcached

ElastiCache a versão 1.5.16 para Memcached adiciona suporte para a versão 1.5.16 do Memcached. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas de [Memcached 1.5.14](#) e [Memcached 1.5.15](#).

Para obter mais informações, consulte as [notas de lançamento do Memcached 1.5.16](#) em Memcached on. GitHub

ElastiCache versão 1.5.10 para Memcached

ElastiCache A versão 1.5.10 para Memcached oferece suporte aos seguintes recursos do Memcached:

- Rebalanceamento de slab automatizado.
- Pesquisas de tabela de hash mais rápidas com o algoritmo murmur3.
- Algoritmo LRU segmentado.
- Crawler de LRU para memória de recuperação do plano de fundo.
- `--enable-seccomp`: uma opção de tempo de compilação.

Ele também apresenta os parâmetros `no_modern` e `inline_ascii_resp`. Para obter mais informações, consulte [Alterações de parâmetros do Memcached 1.5.10](#).

Os aprimoramentos do Memcached adicionados desde a ElastiCache versão 1.4.34 para o Memcached incluem o seguinte:

- Correções cumulativas, como os multiget ASCII, CVE-2017-9951 e limites de crawls para o `metadumper`.
- Melhor gerenciamento de conexão ao fechar as conexões no limite da conexão.
- Gerenciamento do tamanho do item aprimorado para tamanho do item acima de 1MB.
- Melhor desempenho e melhorias de sobrecarga da memória reduzindo os requisitos de memória por item por alguns bytes.

Para obter mais informações, consulte as [notas de lançamento do Memcached 1.5.10](#) em Memcached on. GitHub

ElastiCache versão 1.4.34 para Memcached

ElastiCache a versão 1.4.34 para Memcached não adiciona novos recursos à versão 1.4.33. A versão 1.4.34 é uma versão da correção de bugs maior que a versão usual.

Para obter mais informações, consulte as [notas de lançamento do Memcached 1.4.34](#) em Memcached on. GitHub

ElastiCache versão 1.4.33 para Memcached

Melhorias adicionadas desde a versão 1.4.24 incluem o seguinte:

- Capacidade de despejar todos os metadados de uma classe de slab específica, uma lista de classes de slab ou todas as classes de slab. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.31](#).
- Melhor suporte para itens grandes acima do padrão de 1 megabyte. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.29](#).
- Capacidade de especificar por quanto tempo um cliente pode estar ocioso antes de ser solicitado a fechar.

Capacidade de aumentar dinamicamente a quantidade de memória disponível para o Memcached sem ter que reiniciar o cluster. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.27](#).

- Agora há suporte para o registro em log de fetchers, mutations e evictions. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.26](#).
- A memória liberada pode ser recuperada para um grupo global e reatribuída a novas classes de slab. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.25](#).
- Diversas correções de bugs.
- Alguns novos comandos e parâmetros. Para ver uma lista, consulte [Parâmetros adicionados do Memcached 1.4.33](#).

ElastiCache versão 1.4.24 para Memcached

Melhorias adicionadas desde a versão 1.4.14 incluem o seguinte:

- Gerenciamento de LRUs (menos utilizados recentemente) usando um processo em segundo plano.
- Adição da opção de usar o jenkins ou o murmur3 como algoritmo de hash.
- Alguns novos comandos e parâmetros. Para ver uma lista, consulte [Parâmetros adicionados do Memcached 1.4.24](#).
- Diversas correções de bugs.

ElastiCache versão 1.4.14 para Memcached

Melhorias adicionadas desde a versão 1.4.5 incluem o seguinte:

- Maior capacidade de rebalanceamento de slabs.

- Melhoria de desempenho e escalabilidade.
- Introdução do comando touch para atualizar o tempo de expiração de um item existente sem buscá-lo.
- Descoberta automática: a capacidade de programas clientes determinarem automaticamente todos os nós do cache em um cluster e iniciarem e manterem conexões com todos esses nós.

ElastiCache versão 1.4.5 para Memcached

ElastiCache a versão 1.4.5 para Memcached foi o mecanismo inicial e a versão suportados pela Amazon ElastiCache para Memcached.

Principais diferenças de comportamento e compatibilidade do mecanismo da versão com o Valkey

O Valkey 7.2.6 tem diferenças de compatibilidade semelhantes às versões anteriores do Redis OSS 7.2.4. Para a versão compatível mais recente do Valkey, consulte [Mecanismos e versões compatíveis](#).

Para ter mais informações sobre o Valkey versão 7.2, consulte [Redis OSS 7.2.4 Release Notes](#) (o Valkey 7.2 inclui todas as alterações do Redis OSS até a versão 7.2.4) e [Valkey 7.2 release notes](#) no repositório do Valkey no GitHub.

Aqui estão as mudanças de comportamento que podem causar interrupção entre o Valkey 7.2 e o Redis OSS 7.1 (ou 7.0):

- A amostragem de tempo de congelamento ocorre durante a execução do comando e nos scripts.
- Um comando de fluxo bloqueado que é lançado quando a chave não existe mais carrega um código de erro diferente (-NOGROUP ou -WRONGTYPE em vez de -UNBLOCKED).
- O rastreamento do lado do cliente para scripts agora rastreia as chaves que são lidas pelo script, em vez das chaves declaradas pelo chamador de EVAL/FCALL.

Principais diferenças de comportamento e compatibilidade do mecanismo da versão com o Redis OSS

Important

A página a seguir é estruturada para indicar todas as diferenças de incompatibilidade entre versões e informar você sobre quaisquer considerações que deva fazer ao atualizar para versões mais recentes. Essa lista inclui todos os problemas de incompatibilidade de versão que você possa encontrar ao atualizar.

Você pode fazer o upgrade diretamente da sua versão atual do Redis OSS para a versão mais recente disponível, sem a necessidade de atualizações sequenciais. Por exemplo, você pode atualizar diretamente da versão 3.0 do Redis OSS para a versão 7.0.

As versões do Redis OSS são identificadas com uma versão semântica que compreende um componente MAJOR, MINOR e PATCH. Por exemplo, no Redis OSS 4.0.10, a versão principal é a 4,

a versão secundária é a 0 e a versão do patch é a 10. Esses valores geralmente são incrementados com base nas seguintes convenções:

- As versões principais são para alterações incompatíveis com a API
- As versões secundárias são para novas funcionalidades adicionadas de forma compatível com as versões anteriores
- As versões do patch são para correções de bugs compatíveis com versões anteriores e mudanças não funcionais

Recomendamos sempre permanecer na versão mais recente do patch dentro de uma determinada versão major.minor para obter as melhorias mais recentes de performance e estabilidade. A partir da ElastiCache versão 6.0 para Redis OSS, ElastiCache oferecerá uma única versão para cada versão secundária do Redis OSS, em vez de oferecer várias versões de patch. ElastiCachegerenciará automaticamente a versão de patch de seus clusters em execução, garantindo melhor desempenho e segurança aprimorada.

Também recomendamos atualizar periodicamente para a versão principal mais recente, já que a maioria das melhorias principais não são transferidas para versões mais antigas. À medida que ElastiCache expande a disponibilidade para uma nova AWS região, o OSS ElastiCache para Redis oferece suporte às duas versões major.minor mais recentes da época para a nova região. Por exemplo, se uma nova AWS região for lançada e as ElastiCache versões major.minor mais recentes do Redis OSS forem 7.0 e 6.2, ElastiCache oferecerá suporte às versões 7.0 e 6.2 do Redis OSS na nova região.AWSÀ medida que novas versões major.minor do ElastiCache for Redis OSS forem lançadas, ElastiCache continuaremos adicionando suporte às versões recém-lançadas. Para saber mais sobre como escolher regiões para ElastiCache, consulte [Escolha de regiões e zonas de disponibilidade](#).

Ao fazer uma atualização que abrange versões principais ou secundárias, considere a lista a seguir, que inclui comportamento e alterações incompatíveis com versões anteriores lançadas com o Redis OSS ao longo do tempo.

Alterações de comportamento do Redis OSS 7.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 7.0](#).

- `SCRIPT LOAD` e `SCRIPT FLUSH` não são mais propagados para réplicas. Se você precisar ter alguma durabilidade para scripts, recomendamos que você considere usar as [funções do Redis OSS](#).
- Os canais Pubsub agora estão bloqueados por padrão para novos usuários de ACL.
- O comando `STRALGO` foi substituído pelo comando `LCS`.
- O formato de ACL `GETUSER` foi alterado para que todos os campos mostrem o padrão de cadeia de caracteres de acesso padrão. Se você teve automação usando ACL `GETUSER`, verifique se ela funcionará com qualquer um dos formatos.
- As categorias de ACL para `SELECT`, `WAIT`, `ROLE`, `LASTSAVE`, `READONLY`, `READWRITE` e `ASKING` foram alteradas.
- O comando `INFO` agora mostra estatísticas de comando por subcomando, em vez de nos comandos do contêiner de nível superior.
- Os valores de retorno dos comandos `LPOP`, `RPOP`, `ZPOPMIN` e `ZPOPMAX` foram alterados em determinados casos de borda. Se você usar esses comandos, verifique as notas de release e avalie se foi afetado.
- Os comandos `SORT` e `SORT_R0` agora exigem acesso a todo o espaço de teclas para usar os argumentos `GET` e `BY`.

Alterações de comportamento do Redis OSS 6.2 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 6.2](#).

- Os sinalizadores ACL dos comandos `TIME`, `ECHO`, `ROLE` e `LASTSAVE` foram alterados. Isso pode fazer com que comandos que antes tinham permissão sejam rejeitados e vice-versa.

Note

Nenhum desses comandos modifica ou concede acesso aos dados.

- Ao atualizar do Redis OSS 6.0, a ordem dos key/value pares retornados de uma resposta do mapa para um script lua é alterada. Se os seus scripts usarem `redis.setresp()` ou retornarem um mapa (novo no Redis OSS 6.0), considere as implicações que o script pode quebrar nas atualizações.

Alterações de comportamento do Redis OSS 6.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 6.0](#).

- O número máximo de bancos de dados permitidos foi reduzido de 1,2 milhão para 10 mil. O valor padrão é 16, e desencorajamos o uso de valores muito maiores do que isso, pois encontramos problemas de performance e memória.
- Defina o `AutoMinorVersionUpgrade` parâmetro como `sim` e ElastiCache gerenciará a atualização da versão secundária por meio de atualizações de autoatendimento. Isso será tratado por meio de canais padrão de notificação do cliente por meio de uma campanha de atualização de autoatendimento. Para obter mais informações, consulte [Atualizações de autoatendimento em ElastiCache](#).

Alterações de comportamento do Redis OSS 5.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 5.0](#).

- Os scripts são replicados por efeitos em vez de reexecutar o script na réplica. Isso geralmente melhora a performance, mas pode aumentar a quantidade de dados replicados entre os primários e as réplicas. Há uma opção para voltar ao comportamento anterior que só está disponível na ElastiCache versão 5.0 para Redis OSS.
- Se você estiver atualizando a partir do Redis OSS 4.0, alguns comandos em scripts LUA retornarão argumentos em uma ordem diferente daquela em versões anteriores. No Redis OSS 4.0, o Redis OSS ordenaria algumas respostas lexicograficamente para tornar as respostas determinísticas, essa ordem não é aplicada quando os scripts são replicados por efeitos.
- No Redis OSS 5.0.3 e superior, ElastiCache para o Redis, o OSS descarregará parte do trabalho de E/S para núcleos em segundo plano em tipos de instância com mais de 4 VCPUs. Isso pode alterar as características de performance do Redis OSS e alterar os valores de algumas métricas. Para obter mais informações, consulte [Que métricas devo monitorar?](#) para entender se você precisa alterar quais métricas você assiste.

Alterações de comportamento do Redis OSS 4.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 4.0](#).

- O log lento agora registra dois argumentos adicionais, o nome e o endereço do cliente. Essa alteração deve ser compatível com versões anteriores, a menos que você confie explicitamente em cada entrada de log lenta contendo 3 valores.
- O `CLUSTER NODES` agora retorna um formato ligeiramente diferente, que não é compatível com versões anteriores. Recomendamos que os clientes não usem esse comando para aprender sobre os nós presentes em um cluster e, em vez disso, eles usem `CLUSTER SLOTS`.

EOL passado

Alterações de comportamento do Redis OSS 3.2 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 3.2](#).

- Não há mudanças de compatibilidade a serem destacadas para esta versão.

Para obter mais informações, consulte [ElastiCache versões para o cronograma de fim de vida útil do Redis OSS](#).

Alterações de comportamento do Redis OSS 2.8 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis OSS 2.8](#).

- A partir do Redis OSS 2.8.22, o Redis OSS AOF não é mais suportado no Redis OSS. ElastiCache Recomendamos o uso do MemoryDB quando os dados precisam ser persistidos de forma duradoura.
- A partir do Redis OSS 2.8.22, ElastiCache para Redis, o OSS não suporta mais a anexação de réplicas às primárias hospedadas nele. ElastiCache Durante a atualização, as réplicas externas serão desconectadas e não poderão se reconectar. Recomendamos usar o cache do lado do cliente, disponibilizado no Redis OSS 6.0 como uma alternativa às réplicas externas.
- Os comandos `TTL` e `PTTL` agora retornarão -2 se a chave não existir e -1 se existir, mas não tiver expiração associada. O Redis OSS 2.6 e as versões anteriores costumavam retornar -1 para ambas as condições.

- O SORT com ALPHA agora classificará de acordo com o local de agrupamento local, se nenhuma opção STORE for usada.

Para obter mais informações, consulte [ElastiCache versões para o cronograma de fim de vida útil do Redis OSS](#).

Considerações sobre atualização ao trabalhar com clusters baseados em nós

Note

As considerações a seguir só se aplicam ao atualizar clusters baseados em nós. Eles não se aplicam ao ElastiCache Serverless.

Considerações sobre Valkey e Redis OSS

Ao atualizar um cluster baseado em nós, leve em consideração o seguinte.

- O gerenciamento da versão do mecanismo foi desenvolvido para que você possa ter o máximo controle possível sobre a execução de patches. No entanto, ElastiCache se reserva o direito de corrigir seu cluster em seu nome no caso improvável de uma vulnerabilidade crítica de segurança no sistema ou no software de cache.
- Começando com a ElastiCache versão 7.2 para Valkey e a ElastiCache versão 6.0 para Redis OSS, ElastiCache oferecerá uma única versão para cada versão secundária, em vez de oferecer várias versões de patch.
- A partir do mecanismo Redis OSS versão 5.0.6, você pode atualizar a versão do cluster com o mínimo de tempo de inatividade. O cluster estará disponível para leituras durante todo o processo de atualização e para gravações durante a maior parte da atualização, exceto durante a operação de failover que dura alguns segundos.
- Você também pode atualizar seus ElastiCache clusters com versões anteriores à 5.0.6. O processo envolvido é o mesmo, mas pode incorrer em tempo de failover mais longo durante a propagação do DNS (30 s a 1 m).
- A partir do Redis OSS 7, ElastiCache oferece suporte à alternância entre Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado).

- O processo de atualização do mecanismo Amazon ElastiCache for Redis OSS foi projetado para fazer o melhor esforço para reter seus dados existentes e requer uma replicação bem-sucedida do Redis OSS.
- Ao atualizar o mecanismo, ElastiCache encerrará as conexões existentes do cliente. Para minimizar o tempo de inatividade durante as atualizações do mecanismo, recomendamos implementar [as práticas recomendadas para clientes do Redis OSS](#) com novas tentativas de erro e recuo exponencial e as práticas recomendadas para [minimizar o tempo de inatividade durante a manutenção](#).
- Não é possível atualizar diretamente do Valkey ou Redis OSS (modo cluster desabilitado) para o Valkey ou Redis OSS (modo cluster habilitado) ao atualizar seu mecanismo. O procedimento a seguir mostra como atualizar do Valkey ou Redis OSS (modo cluster desabilitado) para o Valkey ou Redis OSS (modo cluster habilitado).

Para atualizar de uma versão de mecanismo Valkey ou Redis OSS (modo cluster desabilitado) para Valkey ou Redis OSS (modo cluster habilitado)

1. Faça um backup do seu cluster do Valkey ou Redis OSS (modo cluster desabilitado) ou do grupo de replicação. Para obter mais informações, consulte [Realização de backups manuais](#).
 2. Use o backup para criar e propagar um cluster do Valkey ou Redis OSS (modo cluster habilitado) com um fragmento (grupo de nós). Especifique a nova versão do mecanismo e habilite o modo de cluster ao criar o cluster ou o grupo de replicação. Para obter mais informações, consulte [Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente](#).
 3. Exclua o antigo cluster do Valkey ou Redis OSS (modo cluster desabilitado) ou o grupo de replicação. Para acessar mais informações, consulte [Excluindo um cluster no ElastiCache](#) ou [Exclusão de um grupo de replicação](#).
 4. Escale o novo cluster ou grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) para o número de fragmentos (grupos de nós) que você precisa. Para obter mais informações, consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).
- Ao atualizar as principais versões do mecanismo, por exemplo, de 5.0.6 para 6.0, também é necessário selecionar um novo grupo de parâmetros que seja compatível com a nova versão do mecanismo.
 - Para clusters Redis OSS de nó único e clusters Redis individuais com o Multi-AZ desabilitado, recomendamos que seja disponibilizada memória suficiente para o Redis OSS, conforme descrito em [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#).

Nesses casos, o primário não está disponível para solicitações de serviço durante o processo de atualização.

- Para clusters Redis OSS com o Multi-AZ habilitado, também recomendamos que você programe atualizações do mecanismo durante períodos de baixo tráfego de gravações recebidas. Ao atualizar para o Redis OSS 5.0.6 ou posterior, o cluster primário continua disponível para solicitações de serviço durante o processo de atualização.

Os clusters e os grupos de replicação com vários fragmentos são processados e corrigidos da seguinte forma:

- Todos os estilhaços são processados em paralelo. Somente uma operação de atualização é realizada em um estilhaço por vez.
- Em cada fragmento, todas as réplicas são processadas antes do processamento da primária. Caso haja menos réplicas em um fragmento, a primária nesse fragmento pode ser processada antes da conclusão do processamento das réplicas em outros fragmentos.
- Em todos os fragmentos, os nós primários são processados em série. Somente um nó primário é atualizado por vez.
- Caso a criptografia esteja habilitada no cluster ou no grupo de replicação atual, não será possível atualizar para uma versão de mecanismo que não ofereça suporte à criptografia, como de 3.2.6 a 3.2.10.

Considerações sobre o Memcached

Ao atualizar um cluster Memcached baseado em nós, considere o seguinte.

- O gerenciamento da versão do mecanismo foi desenvolvido para que você possa ter o máximo controle possível sobre a execução de patches. No entanto, ElastiCache se reserva o direito de corrigir seu cluster em seu nome no caso improvável de uma vulnerabilidade crítica de segurança no sistema ou no software de cache.
- Como o mecanismo Memcached não oferece suporte para persistência, as atualizações de versão do mecanismo Memcached são sempre um processo disruptivo que limpa todos os dados do cache no cluster.

Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache

Abaixo, é possível encontrar práticas recomendadas para o Amazon ElastiCache. Seguir essas práticas melhora o desempenho e aumenta a confiabilidade do cache.

Tópicos

- [Práticas recomendadas gerais](#)
- [Práticas recomendadas para usar réplicas de leitura](#)
- [Comandos Valkey, Memcached e Redis OSS compatíveis e restritos](#)
- [Configuração e limites do Valkey e Redis OSS](#)
- [Exemplos de clientes IPv6 para Valkey, Memcached e Redis OSS](#)
- [Práticas recomendadas para clientes \(Valkey e Redis OSS\)](#)
- [Práticas recomendadas para clientes \(Memcached\)](#)
- [Clusters do ElastiCache de pilha dupla habilitados para TLS](#)
- [Gerenciamento de memória reservada para Valkey e Redis OSS](#)
- [Práticas recomendadas ao trabalhar com clusters baseados em nós do Valkey e Redis OSS](#)
- [Estratégias de armazenamento em cache para Memcached](#)

Práticas recomendadas gerais

Abaixo, você pode encontrar informações sobre as melhores práticas para usar as interfaces Valkey, Memcached e Redis OSS. ElastiCache

- Use configurações habilitadas para o modo de cluster — o modo de cluster ativado permite que o cache seja dimensionado horizontalmente para obter maior armazenamento e taxa de transferência do que uma configuração desativada no modo de cluster. ElastiCache serverless só está disponível em uma configuração habilitada para o modo de cluster.
- Usar conexões de longa duração: a criação de uma nova conexão é cara e consome tempo e recursos de CPU do cache. Reutilize conexões quando possível (por exemplo, com agrupamento de conexões) para amortizar esse custo em muitos comandos.
- Leia a partir de réplicas — se você estiver usando réplicas de leitura ElastiCache sem servidor ou provisionadas (clusters baseados em nós), direcione as leituras para as réplicas para obter melhor

escalabilidade e menor latência. and/or As leituras de réplicas acabarão sendo consistentes com a primária.

Em um cluster baseado em nós, evite direcionar solicitações de leitura para uma única réplica de leitura, pois as leituras talvez não estejam temporariamente disponíveis em caso de falha no nó. Configure o cliente para direcionar solicitações de leitura para pelo menos duas réplicas de leitura ou direcione as leituras para uma única réplica e a primária.

ElastiCache Sem servidor, a leitura da porta de réplica (6380) direcionará as leituras para a zona de disponibilidade local do cliente quando possível, reduzindo a latência de recuperação. Ele vai acabar fazendo fallback automaticamente para os outros nós durante as falhas.

- Evite comandos caros — Evite executar operações computacionais e I/O intensivas, como os comandos KEYS e. SMEMBERS Sugerimos essa abordagem porque essas operações aumentam a carga no cluster e geram impacto no desempenho do cluster. Em vez disso, use os comandos SCAN e SSCAN.
- Siga as práticas recomendadas do Lua: evite scripts Lua de longa execução e sempre declare antecipadamente as chaves usadas em scripts Lua. Recomendamos essa abordagem para determinar se o script Lua não está usando comandos entre slots. Certifique-se de que as chaves usadas em scripts Lua pertencem ao mesmo slot.
- Use pub/sub fragmentado — Ao usar o Valkey ou o Redis OSS para oferecer suporte a pub/sub cargas de trabalho com alta taxa de transferência, recomendamos que você use o [pub/sub fragmentado](#) (disponível com o Valkey e com o Redis OSS 7 ou posterior). Os clusters tradicionais habilitados para o modo de cluster pub/sub transmitem mensagens para todos os nós do cluster, o que pode resultar em alta. EngineCPUUtilization Observe isso em ElastiCache comandos tradicionais pub/sub commands internally use sharded pub/sub sem servidor.

Práticas recomendadas para usar réplicas de leitura

Muitos aplicativos, como armazenamentos de sessões, tabelas de classificação e mecanismos de recomendação, exigem alta disponibilidade e processam uma quantidade significativamente maior de operações de leitura do que de operações de gravação. Em geral, esses aplicativos podem tolerar dados um pouco obsoletos (consistência eventual). Isso significa que é aceitável que usuários diferentes vejam, por alguns instantes, versões ligeiramente diferentes dos mesmos dados. Por exemplo:

- Os resultados das consultas em cache geralmente podem tolerar dados um pouco obsoletos, especialmente para padrões de armazenamento em cache, nos quais a fonte da verdade é externa.
- Em uma tabela de classificação de jogos, alguns segundos de atraso nas pontuações atualizadas geralmente não afetam significativamente a experiência do usuário.
- Para armazenamentos de sessões, alguns pequenos atrasos na propagação dos dados da sessão entre réplicas raramente afetam a funcionalidade do aplicativo.
- Os mecanismos de recomendação costumam usar análise de dados históricos, portanto, a consistência em tempo real é menos crítica.

A consistência eventual significa que todos os nós de réplica eventualmente retornarão os mesmos dados quando o processo de replicação for concluído, normalmente em milissegundos. Para esses casos de uso, implementar réplicas de leitura é uma estratégia eficaz para reduzir a latência ao ler da sua ElastiCache instância.

O uso de réplicas de leitura na Amazon ElastiCache pode oferecer benefícios significativos de desempenho por meio de:

Escalabilidade de leitura aprimorada

- Distribui as operações de leitura em vários nós de réplica
- Descarrega o tráfego de leitura do nó primário
- Reduz a latência de leitura ao atender solicitações de réplicas geograficamente mais próximas

Desempenho otimizado do nó primário

- Dedicar recursos do nó primário às operações de gravação
- Reduz a sobrecarga de conexão no nó primário
- Melhora o desempenho de gravação e mantém melhores tempos de resposta durante os períodos de pico de tráfego

Usando a leitura da réplica sem servidor ElastiCache

ElastiCache O serverless fornece dois endpoints diferentes, para diferentes requisitos de consistência. Os dois endpoints usam o mesmo nome de DNS, mas portas diferentes. Para usar

a read-from-replica porta, você deve autorizar o acesso às duas portas do seu aplicativo cliente [configurando os grupos de segurança e as listas de controle de acesso à rede da sua VPC](#).

Endpoint primário (porta 6379)

- Use para operações que exigem consistência imediata
- Garante a leitura da maioria dos up-to-date dados
- Ideal para transações críticas e operações de gravação
- Necessário para operações de gravação
- Exemplo: `test-12345.serverless.use1.cache.amazonaws.com:6379`

Endpoint otimizado para latência (porta 6380)

- Otimizado para operações de leitura que podem tolerar uma eventual consistência
- Quando possível, o ElastiCache serverless encaminha automaticamente as solicitações de leitura para o nó de réplica na zona de disponibilidade local do cliente. Essa otimização fornece menor latência ao evitar a latência adicional de rede resultante da recuperação de dados de um nó em uma zona de disponibilidade diferente.
- ElastiCache serverless seleciona automaticamente os nós disponíveis em outras zonas se um nó local não estiver disponível
- Exemplo: `test-12345.serverless.use1.cache.amazonaws.com:6380`
- Clientes como Glide e Lettuce detectarão e rotearão automaticamente as leituras para o endpoint com latência otimizada se você fornecer a leitura da configuração da réplica. Se seu cliente não é compatível com a configuração de roteamento (por exemplo, valkey-java e versões mais antigas do jedis), será necessário definir a porta e a configuração do cliente corretas para ler as réplicas.

Conectando-se a réplicas de leitura em ElastiCache Serverless - Valkey and Glide

O trecho de código a seguir mostra como você pode configurar a leitura da réplica para ElastiCache Serverless na biblioteca Valkey glide. Não é preciso especificar a porta para leitura das réplicas, mas sim configurar a configuração de roteamento `ReadFrom.PREFER_REPLICA`.

```
package glide.examples;

import glide.api.GlideClusterClient;
import glide.api.logging.Logger;
import glide.api.models.configuration.GlideClusterClientConfiguration;
```

```
import glide.api.models.configuration.NodeAddress;
import glide.api.models.exceptions.ClosingException;
import glide.api.models.exceptions.ConnectionException;
import glide.api.models.exceptions.TimeoutException;
import glide.api.models.configuration.ReadFrom;

import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ExecutionException;

public class ClusterExample {

    public static void main(String[] args) {
        // Set logger configuration
        Logger.setLoggerConfig(Logger.Level.INFO);

        GlideClusterClient client = null;

        try {
            System.out.println("Connecting to Valkey Glide...");

            // Configure the Glide Client
            GlideClusterClientConfiguration config =
GlideClusterClientConfiguration.builder()
                .address(NodeAddress.builder()
                    .host("your-endpoint")
                    .port(6379)
                    .build())
                .useTLS(true)
                .readFrom(ReadFrom.PREFER_REPLICA)
                .build();

            // Create the GlideClusterClient
            client = GlideClusterClient.createClient(config).get();
            System.out.println("Connected successfully.");

            // Perform SET operation
            CompletableFuture<String> setResponse = client.set("key", "value");
            System.out.println("Set key 'key' to 'value': " + setResponse.get());

            // Perform GET operation
            CompletableFuture<String> getResponse = client.get("key");
            System.out.println("Get response for 'key': " + getResponse.get());

            // Perform PING operation
```



```
        CompletableFuture<String> pingResponse = client.ping();
        System.out.println("PING response: " + pingResponse.get());

    } catch (ClosingException | ConnectionException | TimeoutException |
ExecutionException e) {
        System.err.println("An exception occurred: ");
        e.printStackTrace();
    } catch (InterruptedException e) {
        Thread.currentThread().interrupt();
    } finally {
        // Close the client connection
        if (client != null) {
            try {
                client.close();
                System.out.println("Client connection closed.");
            } catch (ClosingException | ExecutionException e) {
                System.err.println("Error closing client: " + e.getMessage());
            }
        }
    }
}
```

Comandos Valkey, Memcached e Redis OSS compatíveis e restritos

Comandos compatíveis do Valkey e do Redis OSS

Comandos compatíveis do Valkey e do Redis OSS

Os comandos do Valkey e Redis OSS a seguir são compatíveis com caches com tecnologia sem servidor. Além desses comandos, estes [Comandos compatíveis do Valkey e do Redis OSS](#) também são compatíveis.

Para obter informações sobre os comandos do filtro Bloom, consulte [Comandos de filtro Bloom](#)

Comandos de bitmap

- BITCOUNT

Conta o número de bits definidos (contagem de população) em uma string.

[Saiba mais](#)

- BITFIELD

Realiza operações arbitrárias de inteiro de campo de bits em strings.

[Saiba mais](#)

- BITFIELD_R0

Realiza operações arbitrárias de inteiro somente leitura de campo de bits em strings.

[Saiba mais](#)

- BITOP

Realiza operações bit a bit em várias strings e armazena o resultado.

[Saiba mais](#)

- BITPOS

Encontra o primeiro bit definido (1) ou apagado (0) em uma string.

[Saiba mais](#)

- GETBIT

Retorna um valor de bit por deslocamento.

[Saiba mais](#)

- SETBIT

Define ou apaga o bit no deslocamento do valor da string. Cria a chave, caso ela não exista.

[Saiba mais](#)

Comandos de gerenciamento de clusters

- CLUSTER COUNTKEYSINSLOT

Retorna o número de chaves em um slot de hash.

[Saiba mais](#)

- CLUSTER GETKEYSINSLOT

Retorna os nomes de chave em um slot de hash.

[Saiba mais](#)

- CLUSTER INFO

Retorna informações sobre o estado de um nó. Em um cache sem servidor, retorna o estado sobre o único “fragmento” virtual exposto ao cliente.

[Saiba mais](#)

- CLUSTER KEYSLOT

Retorna o slot de hash de uma chave.

[Saiba mais](#)

- CLUSTER MYID

Retorna o ID de um nó. Em um cache sem servidor, retorna o estado sobre o único “fragmento” virtual exposto ao cliente.

[Saiba mais](#)

- CLUSTER NODES

Retorna a configuração do cluster de um nó. Em um cache sem servidor, retorna o estado sobre o único “fragmento” virtual exposto ao cliente.

[Saiba mais](#)

- CLUSTER REPLICAS

Lista os nós de réplica de um nó principal. Em um cache sem servidor, retorna o estado sobre o único “fragmento” virtual exposto ao cliente.

[Saiba mais](#)

- CLUSTER SHARDS

Retorna o mapeamento de slots de cluster para fragmentos. Em um cache sem servidor, retorna o estado sobre o único “fragmento” virtual exposto ao cliente.

[Saiba mais](#)

- CLUSTER SLOTS

Retorna o mapeamento de slots de cluster para nós. Em um cache sem servidor, retorna o estado sobre o único “fragmento” virtual exposto ao cliente.

[Saiba mais](#)

- CLUSTER SLOT-STATS

Permite o rastreamento de métricas por slot para contagem de chaves, utilização da CPU, entrada de bytes de rede e saída de bytes de rede.

[Saiba mais](#)

- READONLY

Permite consultas somente leitura para uma conexão com um nó de réplica do Valkey ou Redis OSS Cluster.

[Saiba mais](#)

- READWRITE

Permite consultas de leitura e gravação para uma conexão com um nó de réplica do Valkey ou Redis OSS Cluster.

[Saiba mais](#)

- SCRIPT SHOW

Retorna o código-fonte original de um script no cache do script.

[Saiba mais](#)

Comandos de gerenciamento da conexão

- AUTH

Autentica a conexão.

[Saiba mais](#)

- CLIENT GETNAME

Retorna o nome da conexão.

[Saiba mais](#)

- CLIENT REPLY

Instrui o servidor se deve responder aos comandos.

[Saiba mais](#)

- CLIENT SETNAME

Define o nome da conexão.

[Saiba mais](#)

- ECHO

Retorna a string indicada.

[Saiba mais](#)

- HELLO

Cumprimenta o servidor do Valkey ou Redis OSS.

[Saiba mais](#)

- PING

Retorna a resposta de atividade do servidor.

[Saiba mais](#)

- QUIT

Fecha a conexão.

[Saiba mais](#)

- RESET

Redefine a conexão.

[Saiba mais](#)

- SELECT

Altera o banco de dados selecionado.

[Saiba mais](#)

Comandos genéricos

- COPY

Copia o valor de uma chave para uma nova chave.

[Saiba mais](#)

- DEL

Exclui uma ou mais chaves.

[Saiba mais](#)

- DUMP

Retorna uma representação serializada do valor armazenado em uma chave.

[Saiba mais](#)

- EXISTS

Determina se existe uma ou mais chaves.

[Saiba mais](#)

- EXPIRE

Define o tempo de expiração de uma chave em segundos.

[Saiba mais](#)

- EXPIREAT

Define o tempo de expiração de uma chave como um carimbo de data e hora Unix.

[Saiba mais](#)

- EXPIRETIME

Retorna o tempo de expiração de uma chave como um carimbo de data e hora Unix.

[Saiba mais](#)

- PERSIST

Remove o tempo de expiração de uma chave.

[Saiba mais](#)

- PEXPIRE

Define o tempo de expiração de uma chave em milissegundos.

[Saiba mais](#)

- PEXPIREAT

Define o tempo de expiração de uma chave como um carimbo de data e hora Unix em milissegundos.

[Saiba mais](#)

- PEXPIRETIME

Retorna o tempo de expiração de uma chave como um carimbo de data e hora Unix em milissegundos.

[Saiba mais](#)

- PTTL

Define o tempo de expiração de uma chave em milissegundos.

[Saiba mais](#)

- RANDOMKEY

Retorna um nome de chave aleatório do banco de dados.

[Saiba mais](#)

- RENAME

Renomeia uma chave e substitui o destino.

[Saiba mais](#)

- RENAMENX

Só renomeia uma chave quando o nome da chave de destino não existe.

[Saiba mais](#)

- RESTORE

Cria uma chave a partir da representação serializada de um valor.

[Saiba mais](#)

- SCAN

Itera os nomes de chaves no banco de dados.

[Saiba mais](#)

- SORT

Classifica os elementos em uma lista, conjunto ou conjunto classificado, armazenando o resultado como opção.

[Saiba mais](#)

- SORT_RO

Retorna os elementos classificados de uma lista, um conjunto ou um conjunto classificado.

[Saiba mais](#)

- TOUCH

Retornará o número de chaves existentes dentre as especificadas depois de atualizar a hora em que foram acessadas pela última vez.

[Saiba mais](#)

- TTL

Retorna o tempo de expiração de uma chave em segundos.

[Saiba mais](#)

- TYPE

Determina o tipo de valor armazenado em uma chave.

[Saiba mais](#)

- UNLINK

Exclui de maneira assíncrona uma ou mais chaves.

[Saiba mais](#)

Comandos geoespaciais

- GEOADD

Adiciona um ou mais membros a um índice geoespacial. A chave será criada se não existir.

[Saiba mais](#)

- GEODIST

Retorna a distância entre dois membros de um índice geoespacial.

[Saiba mais](#)

- GEOHASH

Retorna membros de um índice geoespacial como strings geohash.

[Saiba mais](#)

- GEOPOS

Retorna a longitude e a latitude de membros de um índice geoespacial.

[Saiba mais](#)

- GEORADIUS

Consulta um índice geoespacial para membros a uma distância de uma coordenada e, como opção, armazena o resultado.

[Saiba mais](#)

- GEORADIUS_RO

Retorna membros de um índice geoespacial que estejam a uma distância de uma coordenada.

[Saiba mais](#)

- GEORADIUSBYMEMBER

Consulta um índice geoespacial para membros a uma distância de um membro e, como opção, armazena o resultado.

[Saiba mais](#)

- GEORADIUSBYMEMBER_RO

Retorna membros de um índice geoespacial que estejam a uma distância de um membro.

[Saiba mais](#)

- GEOSEARCH

Consulta um índice geoespacial para membros dentro de uma área de uma caixa ou círculo.

[Saiba mais](#)

- GEOSEARCHSTORE

Consulta um índice geoespacial para membros dentro de uma área de uma caixa ou círculo e, como opção, armazena o resultado.

[Saiba mais](#)

Comandos de hash

- HDEL

Exclui um ou mais campos e os valores de um hash. Exclui o hash se nenhum campo permanecer.

[Saiba mais](#)

- HEXISTS

Determina se existe um campo em um hash.

[Saiba mais](#)

- HGET

Retorna o valor de um campo em um hash.

[Saiba mais](#)

- HGETALL

Retorna todos os campos e valores em um hash.

[Saiba mais](#)

- HINCRBY

Incrementa o valor inteiro de um campo em um hash por um número. Usará 0 como valor inicial se o campo não existir.

[Saiba mais](#)

- HINCRBYFLOAT

Incrementa o valor de ponto flutuante de um campo por um número. Usará 0 como valor inicial se o campo não existir.

[Saiba mais](#)

- HKEYS

Retorna todos os campos em um hash.

[Saiba mais](#)

- HLEN

Retorna o número de chaves em um hash.

[Saiba mais](#)

- HMGET

Retorna os valores de todos os campos em um hash.

[Saiba mais](#)

- HMSET

Define os valores de vários campos.

[Saiba mais](#)

- HRANDFIELD

Retorna um ou mais campos aleatórios de um hash.

[Saiba mais](#)

- HSCAN

Itera campos e valores de um hash.

[Saiba mais](#)

- HSET

Cria ou modifica o valor de um campo em um hash.

[Saiba mais](#)

- HSETNX

Só define o valor de um campo em um hash quando o campo não existe.

[Saiba mais](#)

- HSTRLEN

Retorna o tamanho do valor de um campo.

[Saiba mais](#)

- HVALS

Retorna todos os valores em um hash.

[Saiba mais](#)

HyperLogLog Comandos

- PFADD

Adiciona elementos a uma HyperLogLog chave. Cria a chave, caso ela não exista.

[Saiba mais](#)

- PFCOUNT

Retorna a cardinalidade aproximada do (s) conjunto (s) observado (s) pela (s) HyperLogLog chave (s).

[Saiba mais](#)

- PFMERGE

Mescla um ou mais HyperLogLog valores em uma única chave.

[Saiba mais](#)

Comandos de lista

- BLMOVE

Extraí um elemento de uma lista, o envia para outra lista e o retorna. Bloqueia até que um elemento esteja disponível de outra forma. Exclui a lista se o último elemento tiver sido movido.

[Saiba mais](#)

- BLMPOP

Extraí o primeiro elemento de uma das várias listas. Bloqueia até que um elemento esteja disponível de outra forma. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- BLPPOP

Remove e devolva o primeiro elemento em uma lista. Bloqueia até que um elemento esteja disponível de outra forma. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- BRPOP

Remove e devolva o último elemento em uma lista. Bloqueia até que um elemento esteja disponível de outra forma. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- BRPOPLPUSH

Extraí um elemento de uma lista, o envia para outra lista e o retorna. Bloqueia até que um elemento esteja disponível de outra forma. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- LINDEX

Retorna um elemento de uma lista pelo índice.

[Saiba mais](#)

- LINSERT

Insere um elemento antes ou depois de outro elemento em uma lista.

[Saiba mais](#)

- LLEN

Retorna o tamanho de uma lista.

[Saiba mais](#)

- LMOVE

Retorna um elemento depois de extraí-lo de uma lista e enviá-lo para outra. Exclui a lista se o último elemento tiver sido movido.

[Saiba mais](#)

- LMPOP

Retornará vários elementos de uma lista depois de removê-los. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- LPOP

Retornará os primeiros elementos em uma lista depois de removê-la. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- LPOS

Retorna o índice de elementos correspondentes em uma lista.

[Saiba mais](#)

- LPUSH

Anexa um ou mais elementos a uma lista. Cria a chave, caso ela não exista.

[Saiba mais](#)

- LPUSHX

Só anexa um ou mais elementos a uma lista quando a lista existe.

[Saiba mais](#)

- LRANGE

Retorna um intervalo de elementos de uma lista.

[Saiba mais](#)

- LREM

Remove elementos de uma lista. Exclui a lista se o último elemento tiver sido removido.

[Saiba mais](#)

- LSET

Define o valor de um elemento em uma lista pelo índice.

[Saiba mais](#)

- LTRIM

Remove elementos de ambas as extremidades de uma lista. Exclui a lista se todos os elementos tiverem sido cortados.

[Saiba mais](#)

- RPOP

Retorna e remove os últimos elementos de uma lista. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- RPOPLPUSH

Retorna o último elemento de uma lista depois de removê-lo e enviá-lo para outra lista. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- RPUSH

Acrescenta um ou mais elementos a uma lista. Cria a chave, caso ela não exista.

[Saiba mais](#)

- RPUSHX

Acrescenta um elemento a uma lista somente quando a lista existe.

[Saiba mais](#)

Comandos Pub/Sub

 Note

Os comandos PUBSUB usam internamente o PUBSUB fragmentado, logo, os nomes de canais serão misturados.

- PUBLISH

Publica uma mensagem em um canal.

[Saiba mais](#)

- PUBSUB CHANNELS

Retorna os canais ativos.

[Saiba mais](#)

- PUBSUB NUMSUB

Retorna uma contagem de assinantes nos canais.

[Saiba mais](#)

- PUBSUB SHARDCHANNELS

Retorna os canais de fragmento ativos.

[Saiba mais](#)

- PUBSUB SHARDNUMSUB

Retorna a contagem de assinantes de canais de fragmento.

[Saiba mais](#)

- SPUBLISH

Publicar uma mensagem em um canal de fragmento

[Saiba mais](#)

- SSUBSCRIBE

Recebe mensagens publicadas em canais de fragmento.

[Saiba mais](#)

- SUBSCRIBE

Recebe mensagens publicadas em canais publicados.

[Saiba mais](#)

- SUNSUBSCRIBE

Deixa de receber mensagens publicadas em canais de fragmento.

[Saiba mais](#)

- UNSUBSCRIBE

Deixa de receber mensagens publicadas em canais.

[Saiba mais](#)

Comandos de script

- EVAL

Executa um script Lua do lado do servidor.

[Saiba mais](#)

- EVAL_R0

Executa um script Lua do lado do servidor somente leitura.

[Saiba mais](#)

- EVALSHA

Executa um script Lua do lado do servidor por resumo. SHA1

[Saiba mais](#)

- EVALSHA_R0

Executa um script Lua do lado do servidor somente para leitura por digest. SHA1

[Saiba mais](#)

- SCRIPT EXISTS

Determina se os scripts Lua do lado do servidor existem no cache de scripts.

[Saiba mais](#)

- SCRIPT FLUSH

Atualmente, um cache de scripts no-op é gerenciado pelo serviço.

[Saiba mais](#)

- SCRIPT LOAD

Carrega um script Lua do lado do servidor no cache de scripts.

[Saiba mais](#)

Comandos do gerenciamento de clusters

Note

Ao usar ElastiCache clusters baseados em nós para Valkey e Redis OSS, os comandos flush devem ser enviados a cada primário pelo cliente para liberar todas as chaves. ElastiCache O Serverless for Valkey e Redis OSS funciona de forma diferente, pois abstrai a topologia subjacente do cluster. O resultado é que, no ElastiCache Serverless, FLUSHDB e FLUSHALL

comandos sempre liberarão todas as chaves no cluster. Por esse motivo, os comandos de liberação não podem ser incluídos em uma transação com tecnologia sem servidor.

- ACL CAT

Lista as categorias ACL ou os comandos dentro de uma categoria.

[Saiba mais](#)

- ACL GENPASS

Gera uma senha pseudoaleatória e segura que pode ser usada para identificar usuários ACL.

[Saiba mais](#)

- ACL GETUSER

Lista as regras ACL de um usuário.

[Saiba mais](#)

- ACL LIST

Despeja as regras efetivas em formato de arquivo ACL.

[Saiba mais](#)

- ACL USERS

Lista todos os usuários ACL.

[Saiba mais](#)

- ACL WHOAMI

Retorna o nome de usuário autenticado da conexão atual.

[Saiba mais](#)

- DBSIZE

Retorna o número de chaves no banco de dados selecionado no momento. Não é garantido que essa operação seja atômica em todos os slots.

[Saiba mais](#)

- **COMMAND**

Retorna informações detalhadas sobre todos os comandos.

[Saiba mais](#)

- **COMMAND COUNT**

Retorna uma contagem de comandos.

[Saiba mais](#)

- **COMMAND DOCS**

Retorna informações documentais sobre um, vários ou todos os comandos.

[Saiba mais](#)

- **COMMAND GETKEYS**

Extrai os nomes de chaves de um comando arbitrário.

[Saiba mais](#)

- **COMMAND GETKEYSANDFLAGS**

Extrai os nomes das chaves e os sinalizadores de acesso de um comando arbitrário.

[Saiba mais](#)

- **COMMAND INFO**

Retorna informações sobre um, vários ou todos os comandos.

[Saiba mais](#)

- **COMMAND LIST**

Retorna uma lista de nomes de comando.

[Saiba mais](#)

- **COMMANDLOG**

Um contêiner para comandos de log de comandos.

[Saiba mais](#)

- **COMMANDLOG GET**

Retorna as entradas do log de comandos especificado.

[Saiba mais](#)

- **COMMANDLOG HELP**

Mostre um texto útil sobre os diferentes subcomandos.

[Saiba mais](#)

- **COMMANDLOG LEN**

Retorna o número de entradas no tipo de log de comandos especificado.

[Saiba mais](#)

- **COMMANDLOG RESET**

Limpa todas as entradas do tipo de log de comandos especificado.

[Saiba mais](#)

- **FLUSHALL**

Remove todas as chaves de todos os bancos de dados. Não é garantido que essa operação seja atômica em todos os slots.

[Saiba mais](#)

- **FLUSHDB**

Remove todas as chaves do banco de dados atual. Não é garantido que essa operação seja atômica em todos os slots.

[Saiba mais](#)

- **INFO**

Retorna informações e estatísticas sobre o servidor.

[Saiba mais](#)

- **LOLWUT**

Exibe arte por computador e a versão do Valkey ou Redis OSS.

[Saiba mais](#)

- ROLE

Retorna a função de replicação.

[Saiba mais](#)

- TIME

Retorna a hora do servidor.

[Saiba mais](#)

Comandos de conjunto

- SADD

Adiciona um ou mais membros a um conjunto. Cria a chave, caso ela não exista.

[Saiba mais](#)

- SCARD

Retorna o número de membros em um conjunto.

[Saiba mais](#)

- SDIFF

Retorna a diferença de vários conjuntos.

[Saiba mais](#)

- SDIFFSTORE

Armazena a diferença de vários conjuntos em uma chave.

[Saiba mais](#)

- SINTER

Retorna a interseção de vários conjuntos.

[Saiba mais](#)

- SINTERCARD

Retorna o número de membros da interseção de vários conjuntos.

[Saiba mais](#)

- SINTERSTORE

Armazena a interseção de vários conjuntos em uma chave.

[Saiba mais](#)

- SISMEMBER

Determina se um membro pertence a um conjunto.

[Saiba mais](#)

- SMEMBERS

Retorna todos os membros de um conjunto.

[Saiba mais](#)

- SMISMEMBER

Determina se vários membros pertencem a um conjunto.

[Saiba mais](#)

- SMOVE

Move um membro de um conjunto para outro.

[Saiba mais](#)

- SPOP

Retorna um ou mais membros aleatórios de um conjunto depois de removê-los. Exclui o conjunto se o último membro tiver sido extraído.

[Saiba mais](#)

- SRANDMEMBER

Obtenha um ou vários membros aleatórios de um conjunto

[Saiba mais](#)

- SREM

Remove um ou mais membros de um conjunto. Exclui o conjunto se o último membro tiver sido removido.

[Saiba mais](#)

- SSCAN

Itera membros de um conjunto.

[Saiba mais](#)

- SUNION

Retorna a união de vários conjuntos.

[Saiba mais](#)

- SUNIONSTORE

Armazena a união de vários conjuntos em uma chave.

[Saiba mais](#)

Comandos de conjunto classificado

- BZMPOP

Remove e retorna um membro por pontuação de um ou mais conjuntos classificados. Bloqueia até que um membro esteja disponível de outra forma. Exclui o conjunto classificado se o último elemento tiver sido extraído.

[Saiba mais](#)

- BZPOPMAX

Remove e retorna o membro com a pontuação mais alta de um ou mais conjuntos classificados. Bloqueia até que um membro esteja disponível de outra forma. Exclui o conjunto classificado se o último elemento tiver sido extraído.

[Saiba mais](#)

- BZPOPMIN

Remove e retorna o membro com a pontuação mais baixa de um ou mais conjuntos classificados. Bloqueia até que um membro esteja disponível de outra forma. Exclui o conjunto classificado se o último elemento tiver sido extraído.

[Saiba mais](#)

- ZADD

Adiciona um ou mais membros a um conjunto classificado ou atualiza as pontuações. Cria a chave, caso ela não exista.

[Saiba mais](#)

- ZCARD

Retorna o número de membros em um conjunto classificado.

[Saiba mais](#)

- ZCOUNT

Retorna a contagem de membros em um conjunto classificado que tenham pontuações dentro de um intervalo.

[Saiba mais](#)

- ZDIFF

Retorna a diferença entre vários conjuntos classificados.

[Saiba mais](#)

- ZDIFFSTORE

Armazena a diferença de vários conjuntos classificados em uma chave.

[Saiba mais](#)

- ZINCRBY

Incrementa a pontuação de um membro em um conjunto classificado.

[Saiba mais](#)

- ZINTER

Retorna a interseção de vários conjuntos classificados.

[Saiba mais](#)

- ZINTERCARD

Retorna o número de membros da interseção de vários conjuntos classificados.

[Saiba mais](#)

- ZINTERSTORE

Armazena a interseção de vários conjuntos classificados em uma chave.

[Saiba mais](#)

- ZLEXCOUNT

Retorna o número de membros em um conjunto classificado dentro de um intervalo lexicográfico.

[Saiba mais](#)

- ZMPOP

Retorna os membros com maior ou menor pontuação de um ou mais conjuntos classificados depois de removê-los. Exclui o conjunto classificado se o último membro tiver sido extraído.

[Saiba mais](#)

- ZMSCORE

Retorna a pontuação de um ou mais membros em um conjunto classificado.

[Saiba mais](#)

- ZPOPMAX

Retorna os membros com maior pontuação de um conjunto classificado depois de removê-los. Exclui o conjunto classificado se o último membro tiver sido extraído.

[Saiba mais](#)

- ZPOPMIN

Retorna os membros com menor pontuação de um conjunto classificado depois de removê-los. Exclui o conjunto classificado se o último membro tiver sido extraído.

[Saiba mais](#)

- ZRANDMEMBER

Retorna um ou mais membros aleatórios de um conjunto classificado.

[Saiba mais](#)

- ZRANGE

Retorna membros em um conjunto classificado dentro de um intervalo de índices.

[Saiba mais](#)

- ZRANGEBYLEX

Retorna membros em um conjunto classificado dentro de um intervalo lexicográfico.

[Saiba mais](#)

- ZRANGEBYSCORE

Retorna membros em um conjunto classificado dentro de um intervalo de índices.

[Saiba mais](#)

- ZRANGESTORE

Armazena um intervalo de membros do conjunto classificado em uma chave.

[Saiba mais](#)

- ZRANK

Retorna o índice de um membro em um conjunto classificado ordenado por pontuações crescentes.

[Saiba mais](#)

- ZREM

Remove um ou mais membros de um conjunto classificado. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREMRANGEBYLEX

Remove membros em um conjunto classificado dentro de um intervalo lexicográfico. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREMRANGEBYRANK

Remove membros em um conjunto classificado dentro de um intervalo de índices. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREMRANGEBYSCORE

Remove membros em um conjunto classificado dentro de um intervalo de pontuações. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREVRANGE

Retorna membros em um conjunto classificado dentro de um intervalo de índices em ordem inversa.

[Saiba mais](#)

- ZREVRANGEBYLEX

Retorna membros em um conjunto classificado dentro de um intervalo lexicográfico em ordem inversa.

[Saiba mais](#)

- ZREVRANGEBYSCORE

Retorna membros em um conjunto classificado dentro de um intervalo de classificações em ordem inversa.

[Saiba mais](#)

- ZREVRANK

Retorna o índice de um membro em um conjunto classificado ordenado por pontuações decrescentes.

[Saiba mais](#)

- ZSCAN

Itera membros e pontuações de um conjunto classificado.

[Saiba mais](#)

- ZSCORE

Retorna a pontuação de um membro em um conjunto classificado.

[Saiba mais](#)

- ZUNION

Retorna a união de vários conjuntos classificados.

[Saiba mais](#)

- ZUNIONSTORE

Armazena a união de vários conjuntos classificados em uma chave.

[Saiba mais](#)

Comandos de fluxo

- XACK

Retorna o número de mensagens que foram confirmadas com êxito pelo membro do grupo de consumidores de um fluxo.

[Saiba mais](#)

- XADD

Acrescenta uma nova mensagem a um fluxo. Cria a chave, caso ela não exista.

[Saiba mais](#)

- XAUTOCLAIM

Altera, ou adquire, a propriedade de mensagens em um grupo de consumidores, como se as mensagens tivessem sido entregues como membro do grupo de consumidores.

[Saiba mais](#)

- XCLAIM

Altera, ou adquire, a propriedade de uma mensagem em um grupo de consumidores, como se a mensagem tivesse sido entregue como membro do grupo de consumidores.

[Saiba mais](#)

- XDEL

Retorna o número de mensagens depois de removê-las de um fluxo.

[Saiba mais](#)

- XGROUP CREATE

Cria um grupo de consumidores.

[Saiba mais](#)

- XGROUP CREATECONSUMER

Cria um consumidor em um grupo de consumidores.

[Saiba mais](#)

- XGROUP DELCONSUMER

Exclui um consumidor de um grupo de consumidores.

[Saiba mais](#)

- XGROUP DESTROY

Destrói um grupo de consumidores.

[Saiba mais](#)

- XGROUP SETID

Define o ID entregue mais recentemente de um grupo de consumidores.

[Saiba mais](#)

- XINFO CONSUMERS

Retorna uma lista dos consumidores em um grupo de consumidores.

[Saiba mais](#)

- XINFO GROUPS

Retorna uma lista dos grupos de consumidores de um fluxo.

[Saiba mais](#)

- XINFO STREAM

Retorna informações sobre um fluxo.

[Saiba mais](#)

- XLEN

Retorna o número de mensagens em um fluxo.

[Saiba mais](#)

- XPENDING

Retorna as informações e as entradas da lista de entradas pendentes de um grupo de consumidores do fluxo.

[Saiba mais](#)

- XRANGE

Retorna as mensagens de um fluxo dentro de um intervalo de IDs.

[Saiba mais](#)

- XREAD

Retorna mensagens de vários fluxos com IDs mais do que as solicitadas. Bloqueia até que uma mensagem esteja disponível de outra forma.

[Saiba mais](#)

- XREADGROUP

Retorna mensagens novas ou históricas de um fluxo para um consumidor em um grupo. Bloqueia até que uma mensagem esteja disponível de outra forma.

[Saiba mais](#)

- XREVRANGE

Retorna as mensagens de um fluxo dentro de um intervalo de IDs na ordem inversa.

[Saiba mais](#)

- XTRIM

Exclui mensagens do início de um fluxo.

[Saiba mais](#)

Comandos de string

- APPEND

Acrescenta uma string ao valor de uma chave. Cria a chave, caso ela não exista.

[Saiba mais](#)

- DECR

Diminui o valor inteiro de uma chave em um. Usará 0 como valor inicial se a chave não existir.

[Saiba mais](#)

- DECRBY

Diminui um número do valor inteiro de uma chave. Usará 0 como valor inicial se a chave não existir.

[Saiba mais](#)

- GET

Retorna o valor da string de uma chave.

[Saiba mais](#)

- GETDEL

Retorna o valor da string de uma chave depois de excluir a chave.

[Saiba mais](#)

- GETEX

Retorna o valor da string de uma chave depois de definir o tempo de expiração.

[Saiba mais](#)

- GETRANGE

Retorna uma substring da string armazenada em uma chave.

[Saiba mais](#)

- GETSET

Retorna o valor da string anterior de uma chave depois de defini-la como um novo valor.

[Saiba mais](#)

- INCR

Incrementa o valor inteiro de uma chave em um. Usará 0 como valor inicial se a chave não existir.

[Saiba mais](#)

- INCRBY

Incrementa o valor inteiro de uma chave por um número. Usará 0 como valor inicial se a chave não existir.

[Saiba mais](#)

- INCRBYFLOAT

Incrementa o valor de ponto flutuante de uma chave por um número. Usará 0 como valor inicial se a chave não existir.

[Saiba mais](#)

- LCS

Encontra a substring comum mais longa.

[Saiba mais](#)

- MGET

Retorna atomicamente os valores de string de uma ou mais chaves.

[Saiba mais](#)

- MSET

Cria ou modifica atomicamente os valores de string de uma ou mais chaves.

[Saiba mais](#)

- MSETNX

Só modifica atomicamente os valores de string de uma ou mais chaves quando todas as chaves não existem.

[Saiba mais](#)

- PSETEX

Define o valor de string e o tempo de expiração de uma chave em milissegundos. A chave será criada se não existir.

[Saiba mais](#)

- SET

Define o valor de string de uma chave, ignorando o tipo. A chave será criada se não existir.

[Saiba mais](#)

- SETEX

Define o valor de string e o tempo de expiração de uma chave. Cria a chave, caso ela não exista.

[Saiba mais](#)

- SETNX

Só define o valor de string de uma chave quando a chave não existir.

[Saiba mais](#)

- SETRANGE

Substitui uma parte do valor de uma string por outra por um deslocamento. Cria a chave, caso ela não exista.

[Saiba mais](#)

- STRLEN

Retorna o tamanho do valor de uma string.

[Saiba mais](#)

- SUBSTR

Retorna uma substring de um valor de string.

[Saiba mais](#)

Comandos de transação

- DISCARD

Descarta uma transação.

[Saiba mais](#)

- EXEC

Executa todos os comandos em uma transação.

[Saiba mais](#)

- MULTI

Inicia uma transação.

[Saiba mais](#)

Comandos restritos do Valkey e do Redis OSS

Para oferecer uma experiência de serviço gerenciado, ElastiCache restringe o acesso a determinados comandos específicos do mecanismo de cache que exigem privilégios avançados. Para caches executando o Redis OSS, os seguintes comandos não estão disponíveis:

- `acl setuser`
- `acl load`

- `acl save`
- `acl deluser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster addslotsrange`
- `cluster bumpepoch`
- `cluster delslot`
- `cluster delslotsrange`
- `cluster failover`
- `cluster flushslots`
- `cluster forget`
- `cluster links`
- `cluster meet`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `psync`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Além disso, os seguintes comandos não estão disponíveis para caches sem servidor:

- `acl log`
- `client caching`
- `client getredir`

- `client id`
- `client info`
- `client kill`
- `client list`
- `client no-evict`
- `client pause`
- `client tracking`
- `client trackinginfo`
- `client unblock`
- `client unpause`
- `cluster count-failure-reports`
- `commandlog`
- `commandlog get`
- `commandlog help`
- `commandlog len`
- `commandlog reset`
- `fcall`
- `fcall_ro`
- `function`
- `function delete`
- `function dump`
- `function flush`
- `function help`
- `function kill`
- `function list`
- `function load`
- `function restore`
- `function stats`
- `keys`

- `lastsave`
- `latency`
- `latency doctor`
- `latency graph`
- `latency help`
- `latency histogram`
- `latency history`
- `latency latest`
- `latency reset`
- `memory`
- `memory doctor`
- `memory help`
- `memory malloc-stats`
- `memory purge`
- `memory stats`
- `memory usage`
- `monitor`
- `move`
- `object`
- `object encoding`
- `object freq`
- `object help`
- `object idletime`
- `object refcount`
- `pfdebug`
- `pfselftest`
- `punsubscribe`
- `pubsub numpat`
- `punsubscribe`

- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `wait`

Comandos do Memcached compatíveis

ElastiCache O Serverless for Memcached oferece suporte a todos os [comandos](#) memcached no memcached 1.6 de código aberto, exceto os seguintes:

- Como as conexões cliente exigem TLS, o protocolo UDP não é compatível.
- O protocolo binário não é compatível, pois está oficialmente [obsoleto](#) no memcached 1.6.
- Os comandos GET/GETS são limitados a 16 KB para evitar possíveis ataques de DoS ao servidor com a busca de um grande número de chaves.
- O comando `flush_all` atrasado será rejeitado com `CLIENT_ERROR`.
- Comandos que configuram o mecanismo ou revelam informações internas sobre o estado ou os logs do mecanismo não são compatíveis, como:
 - Para comando `STATS`, só `stats` e `stats reset` são compatíveis. Outras variações vão retornar `ERROR`
 - `lru / lru_crawler`: modificação das configurações de crawler LRU e LRU
 - `watch`: observa os logs de servidor do memcached
 - `verbosity`: configura o nível de log do servidor
 - `me`: o comando meta debug (`me`) não é compatível

Configuração e limites do Valkey e Redis OSS

Cada um dos mecanismos Valkey e Redis OSS fornece vários parâmetros de configuração, alguns dos quais são modificáveis no Redis OSS e outros não são modificáveis ElastiCache para fornecer desempenho e confiabilidade estáveis.

Caches sem servidor

Para caches de tecnologia sem servidor, grupos de parâmetros não são usados e todas as configurações do Valkey e do Redis OSS não são modificáveis. Os seguintes parâmetros do Valkey e do Redis OSS estão em vigor:

Nome	Detalhes	Description
<code>acl-pubsub-default</code>	<code>allchannels</code>	Permissões de canal pubsub padrão para usuários ACL no cache.
<code>client-output-buffer-limit</code>	<code>normal 0 0 0</code> <code>pubsub 32mb 8mb 60</code>	Cientes normais não têm limite de buffer. PUB/SUB os clientes serão desconectados se violarem o backlog de 32 MiB ou violarem o backlog de 8 MiB por 60s.
<code>client-query-buffer-limit</code>	15 GiB	O tamanho máximo de um único buffer de consulta do cliente. Além disso, os clientes não podem emitir uma solicitação com mais de 3.999 argumentos.
<code>cluster-allow-pubsubshard-when-down</code>	yes	Isso permite que o cache ofereça tráfego pubsub enquanto o cache está parcialmente inativo.
<code>cluster-allow-reads-when-down</code>	yes	Isso permite que o cache ofereça tráfego de leitura enquanto o cache está parcialmente inativo.
<code>cluster-enabled</code>	yes	Todos os caches sem servidor estão habilitados para o modo de cluster, o que permite particionar os dados de maneira transparente em vários fragmentos de back-end. Todos os slots são apresentados aos clientes como pertencentes a um único nó virtual.

Nome	Detalhes	Description
<code>cluster-require-full-coverage</code>	no	Quando o espaço de chaves estiver parcialmente inativo (ou seja, pelo menos um slot de hash estiver inacessível), o cache continuará aceitando consultas para a parte do espaço de chaves que ainda está coberta. Todo o espaço de chaves sempre será “coberto” por um único nó virtual em <code>cluster slots</code> .
<code>lua-time-limit</code>	5000	O tempo máximo de execução de um script Lua, em milissegundos, antes de ElastiCache agir para interromper o script. Se <code>lua-time-limit</code> for excedido, todos os comandos do Valkey e do Redis OSS poderão retornar um erro no formato <code>____-BUSY</code>. Como esse estado pode causar interferência em muitas operações essenciais do Valkey ou do Redis OSS, primeiro ElastiCache emitirá um comando <code>SCRIPT KILL</code>. Se isso não for bem-sucedido, ElastiCache reiniciará forçosamente o Valkey ou o Redis OSS.
<code>maxclients</code>	65000	O número máximo de clientes que podem ser conectados ao cache ao mesmo tempo. Mais conexões estabelecidas podem ou não ser bem-sucedidas.
<code>maxmemory-policy</code>	<code>volatile-lru</code>	Os itens com um conjunto TTL são despejados após a estimativa least-recently-used (LRU) quando o limite de memória do cache é atingido.
<code>notify-keyspace-events</code>	(uma string vazia)	No momento, os eventos do espaço de chaves não são compatíveis com caches sem servidor.

Nome	Detalhes	Description
port	Porta primária: 6379 Porta de leitura: 6380	Os caches sem servidor anunciam duas portas com o mesmo nome de host. A porta primária permite gravações e leituras, e a porta de leitura permite leituras finais consistentes de latência mais baixa usando o comando READONLY.
proto-max-bulk-len	50 MiB	O tamanho máximo de uma única solicitação de elemento.
timeout	0	Os clientes não são desconectados de maneira forçada em um tempo de inatividade específico, mas podem ser desconectados durante o estado estacionário para fins de balanceamento de carga.

Além disso, os seguintes limites estão em vigor:

Nome	Detalhes	Description
Tamanho por cache	5.000 GiB	Quantidade máxima de dados que podem ser armazenados por cache com tecnologia sem servidor.
Tamanho por slot	32 GiB	O tamanho máximo de um único slot de hash do Valkey ou Redis OSS. Os clientes que tentarem definir mais dados do que isso em um único slot do Valkey ou Redis OSS vão disparar a política de remoção no slot e, se nenhuma chave for removível, receberão um erro de falta de memória (OOM).
ECPUs por cache	15.000.000 ECPUs/segundo	ElastiCache Métrica de unidades de processamento (ECPUs). O número de solicitações ECPUs

Nome	Detalhes	Description
		consumidas por suas solicitações depende do tempo gasto pela vCPU e da quantidade de dados transferidos.
ECPUs por slot	30K - 90K ECPUs/segundo	Máximo de 30K ECPUs/second por slot ou 90K ECPUs/second ao usar Read from Replica usando conexões READONLY.
Argumentos por solicitação	3.999	Número máximo de argumentos por solicitação. Os clientes que fizerem mais argumentos por solicitação receberão um erro.
Tamanho do nome da chave	8 KiB	O tamanho máximo de uma única chave do Valkey ou do Redis OSS ou nome de canal. Os clientes que fizerem referência a chaves maiores do que isso receberão um erro.
Tamanho do script Lua	50 MiB	O tamanho máximo de um único script Lua do Valkey ou Redis OSS. As tentativas de carregar um script Lua maior do que isso receberão um erro.

Clusters baseados em nós

Para clusters baseados em nós, consulte [Parâmetros do Valkey e do Redis OSS](#) em busca dos valores padrão dos parâmetros de configuração e quais são configuráveis. Os valores padrão costumam ser recomendados, a menos que você tenha um caso de uso específico que exija que eles sejam substituídos.

Exemplos de clientes IPv6 para Valkey, Memcached e Redis OSS

O ElastiCache é compatível com Valkey, Memcached e Redis OSS. Isso significa que os clientes que aceitam conexões IPv6 devem conseguir se conectar a clusters do ElastiCache para Memcached habilitados para IPv6. Há algumas ressalvas que vale a pena observar ao interagir com recursos habilitados para IPv6.

Você pode ver a publicação [Práticas recomendadas para clientes Valkey e Redis](#) no blog de banco de dados da AWS para obter recomendações sobre como configurar clientes Valkey e Redis OSS para recursos do ElastiCache.

Veja a seguir as práticas recomendadas para interagir com recursos do ElastiCache habilitados para IPv6 com bibliotecas de cliente de código aberto comumente usadas.

Clientes validados com Valkey e Redis OSS

O ElastiCache é compatível com o Valkey e com o Redis OSS de código aberto. Isso significa que os clientes Redis OSS de código aberto e Valkey que oferecem suporte a conexões IPv6 devem conseguir se conectar a clusters do ElastiCache para Redis OSS habilitados para IPv6. Além disso, vários dos clientes Python e Java mais populares foram especificamente testados e validados para funcionar com todas as configurações de tipo de rede compatíveis (somente IPv4, somente IPv6 e pilha dupla)

Os clientes a seguir foram validados especificamente para funcionar com todas as configurações de tipo de rede compatíveis com o Valkey e o Redis OSS.

Clientes validados:

- [Redis Py \(\) – 4.1.2](#)
- [Lettuce – Version: 6.1.6.RELEASE](#)
- [Jedis – Versão: 3.6.0](#)

Práticas recomendadas para clientes (Valkey e Redis OSS)

Aprenda as práticas recomendadas para cenários comuns e acompanhe exemplos de código de algumas das bibliotecas de cliente Valkey e Redis OSS de código aberto mais populares (redis-py, PHPRedis e Lettuce), bem como as práticas recomendadas para interagir com recursos do ElastiCache com bibliotecas de cliente Memcached de código aberto comumente usadas.

Tópicos

- [Grande número de conexões \(Valkey e Redis OSS\)](#)
- [Descoberta de cliente do cluster e recuo exponencial \(Valkey e Redis OSS\)](#)
- [Configurar um tempo limite do lado do cliente \(Valkey e Redis OSS\)](#)
- [Configurar um tempo limite de inatividade do lado do servidor \(Valkey e Redis OSS\)](#)

- [Scripts Lua](#)
- [Armazenamento de itens compostos grandes \(Valkey e Redis OSS\)](#)
- [Configuração do cliente do Lettuce \(Valkey e Redis OSS\)](#)
- [Configurar um protocolo preferencial para clusters de pilha dupla \(Valkey e Redis OSS\)](#)

Grande número de conexões (Valkey e Redis OSS)

Caches com tecnologia sem servidor e nós do ElastiCache para Redis OSS individuais dão suporte a até 65 mil conexões de cliente simultâneas. No entanto, para otimizar o desempenho, recomendamos que as aplicações cliente não funcionem de maneira constante nesse nível de conexões. O Valkey e o Redis OSS têm um processo de thread único baseado em um loop de eventos no qual as solicitações de cliente recebidas são processadas sequencialmente. Isso significa que o tempo de resposta de um determinado cliente aumenta à medida que o número de clientes conectados aumenta.

É possível realizar o seguinte conjunto de ações para evitar um gargalo de conexões em um servidor Valkey ou Redis OSS:

- Realize operações de leitura a partir das réplicas de leitura. Isso pode ser feito usando-se os endpoints de leitor do ElastiCache em modo cluster desabilitado ou usando réplicas para leituras em modo de cluster habilitado, inclusive um cache sem servidor.
- Distribua tráfego de gravação em vários nós primários. É possível fazer isso de duas maneiras. É possível usar um cluster multifragmentado do Valkey ou Redis OSS com um cliente compatível com o modo cluster. Também é possível gravar em vários nós primários em modo de cluster desabilitado com fragmentação do lado do cliente. Isso é feito automaticamente em um cache sem servidor.
- Use um pool de conexões quando disponível na biblioteca cliente.

Em geral, criar uma conexão TCP é uma operação cara em termos computacionais em comparação com os comandos típicos do Valkey ou Redis OSS. Por exemplo, lidar com uma solicitação SET/GET é uma ordem de magnitude mais rápida ao reutilizar uma conexão existente. O uso de um pool de conexões de clientes com um tamanho finito reduz a sobrecarga do gerenciamento de conexões. Isso também limita o número de conexões de entrada simultâneas da aplicação cliente.

O seguinte exemplo de código do PHPRedis mostra que uma nova conexão é criada para cada nova solicitação do usuário:

```
$redis = new Redis();  
if ($redis->connect($HOST, $PORT) != TRUE) {  
    //ERROR: connection failed  
    return;  
}  
$redis->set($key, $value);  
unset($redis);  
$redis = NULL;
```

Comparamos esse código em um loop em uma instância do Amazon Elastic Compute Cloud (Amazon EC2) conectada a um nó Graviton2 (m6g.2xlarge) do ElastiCache para o Redis OSS. Colocamos o cliente e o servidor na mesma zona de disponibilidade. A latência média de toda a operação foi de 2,82 milissegundos.

Quando atualizamos o código e usamos conexões persistentes e um pool de conexões, a latência média de toda a operação foi de 0,21 milissegundo:

```
$redis = new Redis();  
if ($redis->pconnect($HOST, $PORT) != TRUE) {  
    // ERROR: connection failed  
    return;  
}  
$redis->set($key, $value);  
unset($redis);  
$redis = NULL;
```

Configurações de redis.ini necessárias:

- redis.pconnect.pooling_enabled=1
- redis.pconnect.connection_limit=10

O seguinte código é um exemplo de um [pool de conexões do Redis-py](#):

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,  
    max_connections=10))  
conn.set(key, value)
```

O seguinte código é um exemplo de um [pool de conexões do Lettuce](#):

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
```

```
GenericObjectPool<StatefulRedisConnection> pool =  
    ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new  
        GenericObjectPoolConfig());  
pool.setMaxTotal(10); // Configure max connections to 10  
try (StatefulRedisConnection connection = pool.borrowObject()) {  
    RedisCommands syncCommands = connection.sync();  
    syncCommands.set(key, value);  
}
```

Descoberta de cliente do cluster e recuo exponencial (Valkey e Redis OSS)

Ao se conectar a um cluster do ElastiCache Valkey ou Redis OSS em modo cluster habilitado, a biblioteca cliente correspondente deve estar ciente do cluster. Os clientes devem obter um mapa dos slots de hash para os nós correspondentes no cluster para enviar solicitações aos nós certos e evitar a sobrecarga de desempenho de processar redirecionamentos de cluster. Dessa forma, o cliente deve descobrir uma lista completa de slots e dos nós mapeados em duas situações diferentes:

- O cliente é inicializado e deve preencher a configuração de slots inicial
- Um redirecionamento MOVED é recebido do servidor, como na situação de um failover, quando todos os slots atendidos pelo nó primário anterior são assumidos pela réplica, ou de refragmentação quando slots são movidos do nó primário de origem para o nó primário de destino

A descoberta de cliente costuma ser feita por meio da emissão de um comando CLUSTER SLOT ou CLUSTER NODE para o servidor Valkey ou Redis OSS. Recomendamos o método CLUSTER SLOT porque ele retorna o conjunto de intervalos de slots e os nós primários e de réplica reassociados ao cliente. Isso não exige análise adicional do cliente e é mais eficiente.

Dependendo da topologia do cluster, o tamanho da resposta para o comando CLUSTER SLOT pode variar com base no tamanho do cluster. Clusters maiores com mais nós produzem uma resposta maior. Assim, é importante garantir que o número de clientes que fazem a descoberta da topologia do cluster não cresça de maneira ilimitada. Por exemplo, quando a aplicação cliente é inicializada ou perde a conexão do servidor e precisa realizar a descoberta de cluster, um erro comum é que a aplicação cliente dispare várias solicitações de reconexão e descoberta sem adicionar um recuo exponencial ao tentar novamente. Isso pode fazer o servidor Valkey ou Redis OSS não responder por um período prolongado, com a utilização da CPU em 100%. A interrupção será prolongada se cada comando CLUSTER SLOT precisar processar um número grande de nós no barramento do cluster. Observamos várias interrupções de cliente por causa desse comportamento em várias linguagens diferentes, inclusive Python (redis-py-cluster) e Java (Lettuce e Redisson).

Em um cache sem servidor, muitos dos problemas são atenuados automaticamente porque a topologia do cluster anunciada é estática e consiste em duas entradas: um endpoint de gravação e um endpoint de leitura. A descoberta de cluster também é distribuída automaticamente por vários nós durante o uso do endpoint de cache. No entanto, as recomendações a seguir continuam sendo úteis.

Para mitigar o impacto causado por um fluxo repentino de solicitações de conexão e descoberta, recomendamos o seguinte:

- Implemente um pool de conexões do cliente com um tamanho finito para limitar o número de conexões de entrada simultâneas da aplicação cliente.
- Quando o cliente se desconectar do servidor por causa do tempo limite, tente novamente com recuo exponencial com instabilidade. Isso ajuda a evitar que vários clientes sobrecarreguem o servidor ao mesmo tempo.
- Use o guia em [Encontrar endpoints de conexão no ElastiCache](#) para encontrar o endpoint do cluster a fim de realizar a descoberta do cluster. Ao fazer isso, você distribui a carga de descoberta em todos os nós do cluster (até 90) em vez de atingir alguns nós propagados codificados no cluster.

Estes são alguns exemplos de código para a lógica de nova tentativa de recuo exponencial em redis-py, phpRedis e Lettuce.

Exemplo 1 da lógica de recuo: redis-py

O redis-py tem um mecanismo de nova tentativa integrado repetido uma vez logo depois de uma falha. Esse mecanismo pode ser habilitado por meio do argumento `retry_on_timeout` fornecido ao criar um objeto do [Redis OSS](#). Aqui, demonstramos um mecanismo de nova tentativa personalizado com recuo exponencial e instabilidade. Enviamos uma solicitação pull para implementar nativamente o recuo exponencial em [redis-py \(#1494\)](#). No futuro, talvez não seja necessário implementar manualmente.

```
def run_with_backoff(function, retries=5):
    base_backoff = 0.1 # base 100ms backoff
    max_backoff = 10 # sleep for maximum 10 seconds
    tries = 0
    while True:
        try:
            return function()
        except (ConnectionError, TimeoutError):
```



```
if tries >= retries:
    raise
    backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
    print(f"sleeping for {backoff:.2f}s")
    sleep(backoff)
    tries += 1
```

É possível acabar usando o seguinte código para definir um valor:

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

Dependendo da workload, talvez você queira alterar o valor de recuo base de 1 segundo para algumas dezenas ou centenas de milissegundos para workloads sensíveis à latência.

Exemplo 2 da lógica de recuo: PHPRedis

O PHPRedis tem um mecanismo de nova tentativa integrado que repete um máximo (não configurável) de dez vezes. Há um atraso configurável entre as tentativas (com uma instabilidade a partir da segunda tentativa). Para obter mais informações, consulte o [código de exemplo](#). Enviamos uma solicitação pull para implementar nativamente o recuo exponencial em [phpRedis \(#1986\)](#), que já foi mesclado e [documentado](#). Para aqueles que estão na versão mais recente do PHPRedis, não será necessário implementar manualmente, mas incluímos aqui a referência para aqueles nas versões anteriores. Por enquanto, este é um exemplo de código que configura o atraso do mecanismo de nova tentativa:

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
    return; // ERROR: connection failed
}
$client->set($key, $value);
```

Exemplo 3 da lógica de recuo: Lettuce

O Lettuce tem mecanismos de nova tentativa integrados com base nas estratégias de recuo exponencial baseado na publicação [Recuo exponencial e instabilidade](#). Este é um trecho de código que mostra a abordagem de instabilidade completa:

```
public static void main(String[] args)
{
    ClientResources resources = null;
    RedisClient client = null;

    try {
        resources = DefaultClientResources.builder()
            .reconnectDelay(Delay.fullJitter(
                Duration.ofMillis(100),    // minimum 100 millisecond delay
                Duration.ofSeconds(5),     // maximum 5 second delay
                100, TimeUnit.MILLISECONDS) // 100 millisecond base
            ).build();

        client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
        client.setOptions(ClientOptions.builder()
            .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
            100 millisecond connection timeout
            .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //
            5 second command timeout
            .build());

        // use the connection pool from above example
    } finally {
        if (connection != null) {
            connection.close();
        }

        if (client != null){
            client.shutdown();
        }

        if (resources != null){
            resources.shutdown();
        }
    }
}
```

Configurar um tempo limite do lado do cliente (Valkey e Redis OSS)

Configuração do tempo limite do lado do cliente

Configure devidamente o tempo limite do lado do cliente a fim de permitir que o servidor tenha tempo suficiente para processar a solicitação e gerar a resposta. Isso também permitirá antecipar-se à falha se a conexão com o servidor não puder ser estabelecida. Determinados comandos do Valkey ou Redis OSS podem ser mais caros do que outros em termos computacionais. Por exemplo, scripts Lua ou transações MULTI/EXEC que contenham vários comandos que devem ser executados de maneira atômica. Em geral, um tempo limite do lado do cliente maior é recomendado para evitar um tempo limite do cliente antes que a resposta seja recebida do servidor, inclusive o seguinte:

- Execução de comandos em várias teclas
- Execução de transações MULTI/EXEC ou scripts Lua que consistem em vários comandos individuais do Valkey ou Redis OSS
- Leitura de valores grandes
- Realização de operações de bloqueio, como BLPOP

No caso de uma operação de bloqueio, como BLPOP, a prática recomendada é definir o tempo limite do comando como um número menor que o tempo limite do soquete.

Estes são exemplos de código para implementar um tempo limite do lado do cliente em redis-py, PHPRedis e Lettuce.

Exemplo 1 de configuração do tempo limite: redis-py

Este é um exemplo de código com redis-py:

```
# connect to Redis server with a 100 millisecond timeout
# give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
print(res)                      # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
                                   # less than the 2 second socket timeout
print(res)
```

Exemplo 2 de configuração do tempo limite: PHPRedis

Este é um exemplo de código com PHPRedis:

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
    return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n";                    // if there is a connection error

$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n";                  // less than the 2 second socket timeout
```

Exemplo 3 de configuração do tempo limite: Lettuce

Este é um exemplo de código com Lettuce:

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
    RedisClient client = null;
    StatefulRedisConnection<String, String> connection = null;
    try {
        client = RedisClient.create(RedisURI.create(HOST, PORT));
        client.setOptions(ClientOptions.builder()
            .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
            100 millisecond connection timeout
            .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
            2 second command timeout
            .build());

        // use the connection pool from above example

        commands.set("key", "value"); // will timeout after 2 seconds
        commands.blpop(1, "list"); // BLP0P with 1 second timeout
    } finally {
        if (connection != null) {
            connection.close();
        }
    }
}
```

```
if (client != null){
    client.shutdown();
}
}
```

Configurar um tempo limite de inatividade do lado do servidor (Valkey e Redis OSS)

Observamos casos nos quais a aplicação de um cliente tem um grande número de clientes ociosos conectados, mas não está enviando comandos ativamente. Nesses cenários, você pode esgotar todas as 65.000 conexões com um grande número de clientes inativos. Para evitar esses cenários, defina a configuração de tempo limite da maneira indicada no servidor por meio de [Parâmetros do Valkey e do Redis OSS](#). Isso garante que o servidor desconecte ativamente os clientes ociosos para evitar um aumento no número de conexões. Essa configuração não está disponível em caches sem servidor.

Scripts Lua

O Valkey e o Redis OSS aceitam mais de 200 comandos, inclusive aqueles para executar scripts Lua. No entanto, quando o assunto são scripts Lua, existem diversas armadilhas que podem afetar a memória e a disponibilidade do Valkey ou do Redis OSS.

Scripts Lua não parametrizados

Cada script Lua é armazenado em cache no servidor Valkey ou Redis OSS antes de ser executado. Os scripts Lua não parametrizados são exclusivos, o que pode fazer com que o servidor Valkey ou Redis OSS armazene um número grande de scripts Lua e consuma mais memória. Para mitigar isso, certifique-se de que todos os scripts Lua estejam parametrizados e execute regularmente o SCRIPT FLUSH para limpar scripts Lua em cache, se necessário.

Também esteja ciente de que as chaves devem ser fornecidas. Se um valor para o parâmetro KEY não for fornecido, o script falhará. Por exemplo, o seguinte não funciona:

```
serverless-test-1st4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return "Hello World"' 0
(error) ERR Lua scripts without any input keys are not supported.
```

Isso funcionará:

```
serverless-test-1st4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return
redis.call("get", KEYS[1])' 1 mykey-2
"myvalue-2"
```

O exemplo a seguir mostra como usar scripts parametrizados: Primeiro, temos um exemplo de uma abordagem não parametrizada que resulta em três scripts Lua em cache diferentes e não é recomendada:

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

Em vez disso, use o seguinte padrão para criar um único script capaz de aceitar parâmetros passados:

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

Scripts Lua de longa duração

Os scripts Lua podem executar vários comandos atomicamente, logo, podem demorar mais para serem concluídos do que um comando normal do Valkey ou Redis OSS. Se o script Lua só executar operações somente leitura, você poderá pará-lo no meio. No entanto, assim que executa uma operação de gravação, o script Lua se torna imortal e deve ser executado até a conclusão. Um script Lua de longa execução que esteja sofrendo mutação pode fazer com que o servidor Valkey ou Redis OSS deixe de responder por um longo tempo. Para mitigar esse problema, evite scripts Lua de longa duração e teste o script em um ambiente de pré-produção.

Script Lua com gravações furtivas

Existem algumas maneiras pelas quais um script Lua pode continuar gravando novos dados no Valkey ou Redis OSS mesmo quando o Valkey ou Redis OSS estiver acima de `maxmemory`:

- O script começa quando o servidor Valkey ou Redis OSS está abaixo de `maxmemory` e contém várias operações de gravação dentro
- O primeiro comando de gravação do script não está consumindo memória (como `DEL`), seguido de mais operações de gravação que consumam memória

- Você pode mitigar esse problema configurando uma política de remoção adequada no servidor Valkey ou Redis OSS diferente de `noeviction`. Isso permite que o Redis OSS remova itens e libere memória entre os scripts Lua.

Armazenamento de itens compostos grandes (Valkey e Redis OSS)

Em alguns cenários, uma aplicação pode armazenar itens compostos grandes no Valkey ou Redis OSS (como um conjunto de dados hash de vários GB). Não se trata de uma prática recomendada porque ela normalmente acarreta problemas de desempenho no Valkey ou Redis OSS. Por exemplo, o cliente pode executar um comando `HGETALL` para recuperar toda a coleção hash de vários GB. Isso pode gerar uma pressão de memória significativa para o servidor Valkey ou Redis OSS, armazenando em buffer o item grande no buffer de saída do cliente. Além disso, para migração de slots em modo de cluster, o ElastiCache não migra slots que contenham itens com tamanho serializado maior que 256 MB.

Para resolver os problemas de itens grandes, temos as seguintes recomendações:

- Divida o item composto grande em vários itens menores. Por exemplo, divida uma coleção de hash grande em campos de valor-chave individuais com um esquema de nome de chave que reflita da maneira indicada a coleção, como o uso de um prefixo comum no nome da chave para identificar a coleção de itens. Se precisar acessar vários campos na mesma coleção atômicamente, você poderá usar o comando `MGET` para recuperar vários valores-chave no mesmo comando.
- Se você tiver avaliado todas as opções e ainda não tiver conseguido dividir o conjunto de dados da coleção grande, tente usar comandos que operem em um subconjunto dos dados na coleção, e não em toda a coleção. Evite ter um caso de uso que exija que você recupere atômicamente toda a coleção de vários GB no mesmo comando. Um exemplo é usar os comandos `HGET` ou `HMGET` em vez de `HGETALL` em coleções de hash.

Configuração do cliente do Lettuce (Valkey e Redis OSS)

Esta seção descreve as opções de configuração recomendadas de Java e Lettuce e como elas se aplicam aos clusters do ElastiCache.

As recomendações nesta seção foram testadas com o Lettuce versão 6.2.2.

Tópicos

- [Exemplo: configuração do Lettuce para modo de cluster, TLS habilitado](#)

- [Exemplo: configuração do Lettuce para modo de cluster desabilitado, TLS habilitado](#)

TTL de cache de DNS do Java

A JVM armazena em cache pesquisas de nome DNS. Ao resolver um nome de host para um endereço IP, a JVM armazena em cache o endereço IP para um período especificado, conhecido como Time-To-Live (TTL – Tempo de duração).

A escolha de um valor TTL envolve um equilíbrio entre latência e capacidade de resposta a mudança. Com TTLs mais curtos, os resolvedores de DNS percebem atualizações no DNS do cluster com maior rapidez. Isso pode fazer com que sua aplicação responda com maior rapidez às substituições ou a outros fluxos de trabalho pelos quais seu cluster passa. No entanto, se o TTL for muito baixo, ele aumentará o volume da consulta, o que pode aumentar a latência de sua aplicação. Embora não exista um valor TTL correto, vale a pena considerar o tempo que você pode esperar para que uma alteração seja implementada ao definir o TTL.

Como os nós do ElastiCache utilizam entradas de nome de DNS que podem mudar, recomendamos configurar a JVM com um valor TTL de cinco a dez segundos. Isso garante que, quando o endereço IP de um nó mudar, a aplicação poderá receber e usar o novo endereço IP do recurso consultando novamente a entrada DNS.

Em algumas configurações do Java, o TTL padrão da JVM é definido de maneira que jamais atualizará entradas DNS até a JVM ser reiniciada.

Para obter detalhes sobre como definir seu TTL da JVM, consulte [Como definir o TTL da JVM](#).

Versão do Lettuce

Recomendamos o Lettuce versão 6.2.2 ou posterior.

Endpoints

Quando você estiver usando clusters habilitados para o modo de cluster, defina o `redisUri` como o endpoint de configuração do cluster. A pesquisa de DNS para esse URI retorna uma lista de todos os nós disponíveis no cluster e é resolvida aleatoriamente para um deles durante a inicialização do cluster. Para obter mais detalhes sobre como a atualização de topologia funciona, consulte `dynamicRefreshResources` mais adiante neste tópico.

SocketOption

Habilite o [KeepAlive](#). Quando habilitada, essa opção reduz a necessidade de lidar com falhas nas conexões durante o runtime de comando.

Defina [Tempo limite de conexão](#) com base nos requisitos da aplicação e na workload. Para ter mais informações, consulte “Tempos limite” posteriormente neste tópico.

ClusterClientOption: opções do cliente habilitadas para o modo de cluster

Habilite [AutoReconnect](#) quando a conexão for perdida.

Defina [CommandTimeout](#). Para obter mais detalhes, consulte a seção “Tempos limite” mais adiante neste tópico.

Defina [nodeFilter](#) para filtrar os nós com falha da topologia. O Lettuce salva todos os nós encontrados na saída “nós de cluster” (como nós com o status PFAIL/FAIL) nas “partições” do cliente (também conhecidas como fragmentos). Durante o processo de criação da topologia do cluster, ele tenta se conectar a todos os nós da partição. Esse comportamento do Lettuce ao adicionar nós com falha pode causar erros de conexão (ou avisos) quando os nós são substituídos por algum motivo.

Por exemplo, depois que um failover é concluído e o cluster inicia o processo de recuperação, enquanto a topologia do cluster é atualizada, o mapa de nós de barramento do cluster tem um curto período em que o nó inativo é listado como um nó com o status FAIL antes de ser completamente removido da topologia. Durante esse período, o cliente do Lettuce o considera um nó íntegro e se conecta de forma contínua a ele. Isso causa uma falha após o término da nova tentativa.

Por exemplo:

```
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        ... // other options
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
            || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
            || it.is(RedisClusterNode.NodeFlag.HANDSHAKE)
            || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();
redisClusterClient.setOptions(clusterClientOptions);
```

 Note

A filtragem de nós é mais bem utilizada com `DynamicRefreshSources` definido como verdadeiro. Do contrário, se a visualização da topologia for obtida de um único nó inicial problemático, que vê um nó primário de algum fragmento com o status de falha, ela filtrará esse nó primário, o que causará a falta de cobertura dos slots. Ter vários nós iniciais (quando `DynamicRefreshSources` é verdadeiro) reduz a probabilidade de ocorrência desse problema, pois pelo menos alguns dos nós iniciais devem ter uma visualização de topologia atualizada após um failover com o primário recém-promovido.

`ClusterTopologyRefreshOptions`: opções para controlar a atualização da topologia do cluster do cliente Habilitado no modo do cluster

 Note

Os clusters com modo de cluster desabilitado não dão suporte a comandos de descoberta do cluster nem são compatíveis com a funcionalidade de descoberta da topologia dinâmica de todos os clientes.

O modo de cluster desabilitado com o ElastiCache não é compatível com `MasterSlaveTopologyRefresh` do Lettuce. Em vez disso, para o modo de cluster desabilitado, você pode configurar um `StaticMasterReplicaTopologyProvider` e fornecer os endpoints de leitura e gravação do cluster.

Para obter mais informações sobre a conexão com clusters no modo de cluster desabilitado, consulte [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#).

Se quiser usar a funcionalidade de descoberta da topologia dinâmica do Lettuce, você poderá criar um cluster com modo de cluster habilitado com a mesma configuração de fragmento do cluster existente. No entanto, para clusters habilitados no modo de cluster, recomendamos configurar pelo menos três fragmentos com no mínimo uma réplica para oferecer compatibilidade com failover rápido.

Habilite [enablePeriodicRefresh](#). Isso possibilita atualizações periódicas da topologia do cluster para que o cliente a atualize nos intervalos do `refreshPeriod` (padrão: 60 segundos). Quando desabilitado, o cliente atualiza a topologia do cluster somente quando ocorrem erros ao tentar executar comandos no cluster.

Com essa opção habilitada, você pode reduzir a latência associada à atualização da topologia do cluster adicionando esse trabalho a uma tarefa em segundo plano. Embora a atualização da topologia seja executada em um trabalho em segundo plano, ela pode ser um pouco lenta para clusters com muitos nós. Isso ocorre porque todos os nós estão sendo consultados para obter a visualização mais atualizada do cluster. Se você executa um cluster grande, é recomendável aumentar o período.

Habilite [enableAllAdaptiveRefreshTriggers](#). Isso possibilita a atualização adaptativa da topologia que usa todos os [acionadores](#): `MOVED_REDIRECT`, `ASK_REDIRECT`, `PERSISTENT_RECONNECTS`, `UNCOVERED_SLOT`, `UNKNOWN_NODE`. Os acionadores de atualização adaptável iniciam atualizações de visualização da topologia com base em eventos que ocorrem durante as operações de cluster do Valkey ou Redis OSS. Quando habilitada, essa opção ocasiona uma atualização imediata da topologia quando ocorre um dos acionadores anteriores. As atualizações adaptáveis acionadas têm uma taxa limitada usando um tempo limite porque os eventos podem ocorrer em grande escala (tempo limite padrão entre as atualizações: 30).

Habilite [closeStaleConnections](#). Isso possibilita fechar conexões obsoletas ao atualizar a topologia do cluster. Ela só entrará em vigor se [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled\(\)](#) for verdadeiro. Quando habilitado, o cliente pode fechar conexões obsoletas e criar outras em segundo plano. Isso reduz a necessidade de lidar com falhas nas conexões durante o runtime de comando.

Habilite [dynamicRefreshResources](#). Recomendamos habilitar a opção `dynamicRefreshResources` para clusters pequenos e desabilitá-la para clusters grandes. A opção `dynamicRefreshResources` possibilita descobrir nós de cluster com base no nó inicial fornecido (por exemplo, endpoint de configuração de cluster). Ele usa todos os nós descobertos como fontes para atualizar a topologia do cluster.

A atualização dinâmica consulta todos os nós descobertos para a topologia do cluster e tenta escolher a visualização mais precisa do cluster. Se for definido como falso, somente os nós iniciais serão usados como fontes para a descoberta da topologia e o número de clientes será obtido apenas para os nós iniciais. Quando desabilitado, se o endpoint de configuração do cluster for resolvido em um nó com falha, a tentativa de atualizar a visualização do cluster falhará e causará exceções. Esse cenário pode ocorrer porque demora até que a entrada de um nó com falha seja removida do endpoint de configuração do cluster. Portanto, o endpoint de configuração ainda pode ser resolvido aleatoriamente em um nó com falha por um curto período.

No entanto, quando está habilitado, usamos todos os nós do cluster recebidos da visualização do cluster para consultar a visualização atual. Como filtramos os nós com falha nessa visualização, a atualização da topologia será bem-sucedida. No entanto, quando `dynamicRefreshSources` é

verdadeiro, o Lettuce consulta todos os nós para obter a visualização do cluster e, depois, compara os resultados. Portanto, isso pode ser caro para clusters com muitos nós. Sugerimos que você desabilite esse recurso para clusters com muitos nós.

```
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();
```

ClientResources

Configure [DnsResolver](#) com [DirContextDnsResolver](#). O resolvidor de DNS é baseado em `com.sun.jndi.dns.DnsContextFactory` do Java.

Configure [reconnectDelay](#) com recuo exponencial e oscilação total. O Lettuce tem mecanismos de repetição integrados com base nas estratégias de recuo exponencial. Para obter detalhes, consulte a publicação [Recuo exponencial e oscilação](#) no Blog de arquitetura da AWS. Para ter mais informações sobre a importância de tentar novamente uma estratégia de recuo, consulte as seções de lógica de recuo da [publicação do blog sobre práticas recomendadas](#) no Blog de bancos de dados da AWS.

```
ClientResources clientResources = DefaultClientResources.builder()
    .dnsResolver(new DirContextDnsResolver())
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),    // minimum 100 millisecond delay
            Duration.ofSeconds(10),    // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .build();
```

Tempos limite

Use um valor de tempo limite de conexão menor do que o tempo limite do comando. O Lettuce utiliza uma conexão preguiçosa. Portanto, se o tempo limite de conexão for maior do que o tempo limite do comando, você poderá ter um período de falha persistente após uma atualização da topologia se o Lettuce tentar se conectar a um nó não íntegro e o tempo limite do comando sempre for excedido.

Utilize um tempo limite de comando dinâmico para comandos diferentes. Recomendamos definir o tempo limite do comando com base na duração esperada do comando. Por exemplo, utilize um

tempo limite maior para comandos que iteram em várias teclas, como scripts FLUSHDB, FLUSHALL, KEYS, SMEMBERS ou Lua. Utilize tempos limite mais curtos para comandos de tecla única, como SET, GET e HSET.

 Note

Os tempos limite configurados no exemplo a seguir são para testes que executaram comandos SET/GET com chaves e valores de até 20 bytes. O tempo de processamento pode ser maior quando os comandos são complexos ou quando as chaves e os valores são maiores. Você deve definir os tempos limite com base no caso de uso de sua aplicação.

```
private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);
```

```
SocketOptions socketOptions = SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .build();
```

```
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
        ImmutableSet.<ProtocolKeyword>builder()
            .add(CommandType.FLUSHDB)
            .add(CommandType.FLUSHALL)
            .add(CommandType.CLUSTER)
            .add(CommandType.INFO)
            .add(CommandType.KEYS)
            .build();

    private final Duration defaultCommandTimeout;
    private final Duration metaCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
```

```

    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(
        new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();

```

Exemplo: configuração do Lettuce para modo de cluster, TLS habilitado

Note

Os tempos limite no exemplo a seguir são para testes que executaram comandos SET/GET com chaves e valores de até 20 bytes. O tempo de processamento pode ser maior quando os comandos são complexos ou quando as chaves e os valores são maiores. Você deve definir os tempos limite com base no caso de uso de sua aplicação.

```

// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
    cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)

```

```

        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),    // minimum 100 millisecond delay
            Duration.ofSeconds(10),    // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
        ImmutableSet.<ProtocolKeyword>builder()
            .add(CommandType.FLUSHDB)
            .add(CommandType.FLUSHALL)
            .add(CommandType.CLUSTER)
            .add(CommandType.INFO)
            .add(CommandType.KEYS)
            .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

```

```
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
        META_COMMAND_TIMEOUT))
    .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();

// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
        .connectTimeout(CONNECT_TIMEOUT)
        .keepAlive(true)
        .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        .topologyRefreshOptions(topologyOptions)
        .socketOptions(socketOptions)
        .autoReconnect(true)
        .timeoutOptions(timeoutOptions)
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
                || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
                || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
```



```
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Exemplo: configuração do Lettuce para modo de cluster desabilitado, TLS habilitado

Note

Os tempos limite no exemplo a seguir são para testes que executaram comandos SET/GET com chaves e valores de até 20 bytes. O tempo de processamento pode ser maior quando os comandos são complexos ou quando as chaves e os valores são maiores. Você deve definir os tempos limite com base no caso de uso de sua aplicação.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
RedisURI redisUriStandalone =

    RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

ClientResources clientResources =
    DefaultClientResources.builder()
        .dnsResolver(new DirContextDnsResolver())
        .reconnectDelay(
            Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(10), // maximum 10 second delay
                100,
                TimeUnit.MILLISECONDS)) // 100 millisecond base
        .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
```

```

    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
ImmutableSet.<ProtocolKeyword>builder()
    .add(CommandType.FLUSHDB)
    .add(CommandType.FLUSHALL)
    .add(CommandType.INFO)
    .add(CommandType.KEYS)
    .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();

final SocketOptions socketOptions =
    SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =

    ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);

```

Configurar um protocolo preferencial para clusters de pilha dupla (Valkey e Redis OSS)

Para clusters do Valkey ou Redis OSS de modo cluster habilitado, você pode controlar o protocolo que os clientes usarão para se conectar aos nós no cluster com o parâmetro de descoberta de IP. O parâmetro de descoberta de IP pode ser definido como IPv4 ou IPv6.

Para clusters do Valkey ou Redis OSS, o parâmetro de descoberta de IP define o protocolo IP usado na saída de [cluster slots \(\)](#), [cluster shards \(\)](#) e [cluster nodes \(\)](#). Esses comandos são usados pelos clientes para descobrir a topologia do cluster. Os clientes usam os IPs nesses comandos para se conectar aos outros nós no cluster.

Alterar a descoberta de IP não resultará em nenhum tempo de inatividade para os clientes conectados. No entanto, as alterações levarão algum tempo para se propagar. Para determinar quando as alterações foram completamente propagadas para um cluster do Valkey ou Redis OSS, monitore a saída de `cluster slots`. Depois que todos os nós retornados pelo comando `cluster slots` relatam IPs com o novo protocolo, as alterações terminam de se propagar.

Exemplo com Redis-Py:

```
cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
nodes):
    nodes = set()

    # This refreshes the cluster topology and will discovery any node updates.
    # Under the hood it calls cluster slots
    cluster.nodes_manager.initialize()
    for node in cluster.get_nodes():
        nodes.add(node.host)
    self.logger.info(nodes)

    time.sleep(1)
```

Exemplo com Lettuce:

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
6379));
```

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

Set<String> nodes;

do {
    // Check for any changes in the cluster topology.
    // Under the hood this calls cluster slots
    clusterClient.refreshPartitions();
    Set<String> nodes = new HashSet<>();

    for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
        nodes.add(node.getUri().getHost());
    }

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
    } catch (UnknownHostException ignored) {}
    return false;
})));
```

Práticas recomendadas para clientes (Memcached)

Conheça as práticas recomendadas para cenários comuns com os clusters do ElastiCache para Memcached.

Tópicos

- [Configurando seu ElastiCache cliente para balanceamento de carga eficiente \(Memcached\)](#)
- [Clientes validados com Memcached](#)
- [Configurar um protocolo preferencial para clusters de pilha dupla \(Memcached\)](#)

Configurando seu ElastiCache cliente para balanceamento de carga eficiente (Memcached)

Note

Esta seção se aplica a clusters Memcached de vários nós baseado em nós.

Para usar com eficiência vários nós do ElastiCache Memcached, você precisa ser capaz de distribuir suas chaves de cache pelos nós. Uma maneira simples de balancear a carga de um cluster com n nós é calcular o hash da chave do objeto e modificar o resultado por n : $\text{hash}(\text{key}) \bmod n$. O valor resultante (de 0 a $n-1$) é o número do nó em que você coloca o objeto.

Essa abordagem é simples e funcionará bem desde que o número de nós (n) seja constante. No entanto, sempre que você adiciona ou remove um nó do cluster, o número de chaves que precisam ser movidas é $(n - 1) / n$ (em que n é o novo número de nós). Assim, essa abordagem resulta em um grande número de chaves movidas, o que se converte em um grande número de perdas no cache inicial, especialmente porque o número de nós fica grande. A escala de 1 para 2 nós resulta na movimentação de $(2-1)/2$ (50%) das chaves, o melhor caso. A escala de 9 a 10 nós resulta na movimentação de $(10-1)/10$ (90%) das chaves. Se você estiver aumentando a escala devido a um pico no tráfego, não convém ter um grande número de perdas no cache. Um grande número de perdas no cache resulta em acertos no banco de dados, que já está sobrecarregado devido ao pico no tráfego.

A solução para esse dilema é o hashing consistente. O hashing consistente usa um algoritmo, de modo que, sempre que um nó é adicionado ou removido a/de um cluster, o número de chaves que devem ser movidas é de aproximadamente $1/n$ (onde n é o novo número de nós). A escala de 1 para 2 nós resulta na movimentação de $1/2$ (50%) das chaves, o pior caso. Escalar de 9 a 10 nós resulta na movimentação de $1/10$ (10%) das chaves.

Como usuário, você controla qual algoritmo de hashing é usado para clusters de vários nós. Recomendamos configurar os clientes para usar o hashing consistente. Felizmente, existem muitas bibliotecas de cliente Memcached na maioria das linguagens populares que implementam o hashing consistente. Verifique a documentação da biblioteca que você está usando para ver se ela oferece suporte para hashing consistente e como implementá-lo.

Se você estiver trabalhando em Java, PHP ou .NET, recomendamos que você use uma das bibliotecas de ElastiCache cliente da Amazon.

Hash consistente usando o Java

O cliente Java ElastiCache Memcached é baseado no cliente Java spymemcached de código aberto, que tem recursos de hashing consistentes incorporados. A biblioteca inclui uma `KetamaConnectionFactory` classe que implementa um hashing consistente. Por padrão, o hashing consistente está desativado em spymemcached.

Para obter mais informações, consulte a `KetamaConnectionFactory` documentação em [KetamaConnectionFactory](#).

Hash consistente usando o PHP com o Memcached

O cliente PHP ElastiCache Memcached é um invólucro da biblioteca PHP integrada do Memcached. Por padrão, o hashing consistente é desativado pela biblioteca PHP Memcached.

Use o código a seguir para ativar o hashing consistente.

```
$m = new Memcached();  
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Além do código anterior, recomendamos que você também ative `memcached.sess_consistent_hash` no seu arquivo `php.ini`.

[Para obter mais informações, consulte a documentação de configuração em tempo de execução do Memcached PHP em http://php.net/manual/en/memcached.configuração.php](http://php.net/manual/en/memcached.configuração.php). Observe especificamente o parâmetro `memcached.sess_consistent_hash`.

Hash consistente usando o .NET com o Memcached

O cliente ElastiCache Memcached .NET é um invólucro do Enyim Memcached. Por padrão, o hashing consistente é ativado pelo cliente Enyim Memcached.

Para obter mais informações, consulte a `memcached/locator` documentação em [https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration# user-content-memcachedlocator](https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator).

Clientes validados com Memcached

Os seguintes clientes foram validados especificamente para funcionar com todas as configurações de tipo de rede aceitas pelo Memcached.

Cientes validados:

- [AWS ElastiCache Cluster Client Memcached for Php, Versão *3.6.2](#)
- [AWS Elasticache Cluster Client Memcached for Java](#) – O mestre mais recente no Github

Configurar um protocolo preferencial para clusters de pilha dupla (Memcached)

Para clusters Memcached, você pode controlar o protocolo que os clientes usarão para se conectar aos nós no cluster com o parâmetro de descoberta de IP. O parâmetro de descoberta de IP pode ser definido como IPv4 ou IPv6.

O parâmetro de descoberta de IP controla o protocolo IP usado na saída do cluster config get. O que, por sua vez, determinará o protocolo IP usado pelos clientes que oferecem suporte à descoberta automática de clusters do ElastiCache para Memcached.

Alterar a descoberta de IP não resultará em nenhum tempo de inatividade para os clientes conectados. No entanto, as alterações levarão algum tempo para se propagar.

Monitore a saída de `getAvailableNodeEndpoints` para Java e para Php monitore a saída `degetServerList`. Depois que a saída dessas funções relata IPs de todos os nós do cluster que usam o protocolo atualizado, as alterações terminam de se propagar.

Exemplo de Java

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

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

Set<String> nodes;

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

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
    } catch (UnknownHostException ignored) {}
    return false;
}));
```

```
}});
```

Exemplo de 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);
```

Qualquer conexão de cliente existente que tenha sido criada antes da atualização da descoberta de IP ainda será conectada usando o protocolo antigo. Todos os clientes validados se reconectarão automaticamente ao cluster usando o novo protocolo IP assim que as alterações forem detectadas na saída dos comandos de descoberta do cluster. No entanto, isso depende da implementação do cliente.

Clusters do ElastiCache de pilha dupla habilitados para TLS

Quando o TLS está habilitado para clusters do ElastiCache, as funções de descoberta de cluster (`cluster slots`, `cluster shards`, e `cluster nodes` para o Redis) ou `config get cluster` para o Memcached retornam nomes de host em vez de IPs. Os nomes de host são então usados em vez de IPs para se conectar ao cluster do ElastiCache e realizar um handshake de TLS. Isso significa que os clientes não serão afetados pelo parâmetro de descoberta de IP. Para clusters habilitados para TLS, o parâmetro de descoberta de IP não tem efeito no protocolo IP preferencial.

Em vez disso, o protocolo IP usado será determinado pelo protocolo IP que o cliente prefere ao resolver nomes de host DNS.

Cientes Java

Ao se conectar de um ambiente Java que oferece suporte a IPv4 e IPv6, o Java, por padrão, prefere IPv4 em vez de IPv6 para compatibilidade com versões anteriores. No entanto, a preferência do protocolo IP é configurável por meio dos argumentos da JVM. Para preferir IPv4, a JVM aceita `-Djava.net.preferIPv4Stack=true` e prefere o conjunto IPv6 - `-Djava.net.preferIPv6Stack=true`. A configuração `-Djava.net.preferIPv4Stack=true` significa que a JVM não fará mais nenhuma conexão IPv6. Para Valkey ou Redis OSS, isso inclui aquelas para outras aplicações não Valkey e não Redis OSS.

Preferências de nível de host

Em geral, se o runtime do cliente ou o cliente não fornecer opções de configuração para definir uma preferência de protocolo IP, ao executar a resolução de DNS, o protocolo IP dependerá da configuração do host. Por padrão, a maioria dos hosts prefere IPv6 em vez de IPv4, mas essa preferência pode ser configurada no nível do host. Isso afetará todas as solicitações de DNS desse host, não apenas aquelas para clusters do ElastiCache.

Hosts Linux

Para Linux, uma preferência de protocolo IP pode ser configurada modificando o arquivo do `gai.conf`. O arquivo do `gai.conf` pode ser encontrado em `/etc/gai.conf`. Se não houver `gai.conf` especificado, um exemplo deve estar disponível em `/usr/share/doc/glibc-common-x.xx/gai.conf`, o qual pode ser copiado em `/etc/gai.conf`, e a configuração padrão não deverá ser comentada. Para atualizar a configuração para preferir IPv4 ao se conectar a um cluster do ElastiCache, atualize a precedência do intervalo CIDR que abrange os IPs do cluster para estar acima da precedência das conexões IPv6 padrão. Por padrão, as conexões IPv6 têm uma precedência de 40. Por exemplo, supondo que o cluster esteja localizado em uma sub-rede com o CIDR `172.31.0.0/16`, a configuração abaixo faria com que os clientes preferissem conexões IPv4 a esse 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

```

Mais detalhes sobre `gai.conf` estão disponíveis na [página principal do Linux](#)

Hosts do Windows

O processo para hosts do Windows é semelhante. Para hosts do Windows, você pode executar `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`. Isso tem o mesmo resultado que modificar o arquivo `gai.conf` em hosts Linux.

Isso atualizará as políticas de preferência para preferir conexões IPv4 em vez de conexões IPv6 para o intervalo CIDR especificado. Por exemplo, supondo que o cluster esteja em uma sub-rede com o CIDR `172.31.0.0:0/16`, a execução de `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` resultaria na seguinte tabela de precedência, o que faria com que os clientes preferissem IPv4 ao se conectarem ao 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
```

Gerenciamento de memória reservada para Valkey e Redis OSS

A memória reservada é a memória reservada para o uso não relacionado a dados. Ao executar um backup ou failover, o Valkey e o Redis OSS usam a memória disponível para registrar operações de gravação no seu cluster enquanto os dados do cluster estão sendo gravados no arquivo .rdb. Se você não tiver memória suficiente disponível para todas as gravações, o processo falhará. A seguir, você encontrará informações sobre opções para gerenciar memória reservada para o ElastiCache para Redis OSS e como aplicar essas opções.

Tópicos

- [De quanta memória reservada você precisa?](#)
- [Parâmetros para gerenciar memória reservada](#)
- [Especificação do parâmetro de gerenciamento de memória reservada](#)

De quanta memória reservada você precisa?

Se você estiver executando uma versão do Redis OSS anterior à 2.8.22, reserve mais memória para backups e failover do que se estiver executando o Redis OSS 2.8.22 ou posterior. Este requisito é devido às diferentes maneiras como o ElastiCache para Redis OSS implementa o processo de backup. A regra geral é reservar metade do valor `maxmemory` de um tipo de nó para a sobrecarga do Redis OSS para versões anteriores à 2.8.22 e um quarto para o Redis OSS versão 2.8.22 e posterior.

Devido às diferentes maneiras pelas quais o ElastiCache implementa o processo de backup e replicação, a regra geral é reservar 25% do valor `maxmemory` de um tipo de nó usando o parâmetro `reserved-memory-percent`. Este é o valor padrão e recomendado para a maioria dos casos.

Quando os tipos de instância micro e pequena com capacidade de intermitência estão operando perto dos limites de `maxmemory`, eles podem experimentar o uso de troca. Para melhorar a confiabilidade operacional desses tipos de instância durante o backup, a replicação e o tráfego intenso, recomendamos aumentar o valor do parâmetro `reserved-memory-percent` em até 30% nos tipos de instâncias pequenas e até 50% nos tipos de instâncias micro.

Para workloads com uso intenso de gravação em clusters do ElastiCache com classificação de dados em níveis, recomendamos aumentar `reserved-memory-percent` para até 50% da memória disponível do nó.

Para obter mais informações, consulte:

- [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#)
- [Como a sincronização e o backup são implementados](#)
- [Classificação de dados em níveis no ElastiCache](#)

Parâmetros para gerenciar memória reservada

Desde 16 de março de 2017, o Amazon ElastiCache fornece dois parâmetros mutuamente exclusivos para gerenciar memória do Valkey ou Redis OSS, `reserved-memory` e `reserved-memory-percent`. Nenhum desses parâmetros faz parte da distribuição do Valkey ou Redis OSS.

Dependendo de quando você se tornou cliente do ElastiCache, um desses dois parâmetros será o parâmetro de gerenciamento de memória padrão. Esse parâmetro se aplica quando você cria um novo grupo de replicação ou cluster do Valkey ou Redis OSS e usa um grupo de parâmetros padrão.

- Para clientes que iniciaram antes de 16 de março de 2017, quando você criar um grupo de replicação ou cluster do Redis OSS usando o grupo de parâmetros padrão, o parâmetro de gerenciamento de memória será `reserved-memory`. Nesse caso, zero (0) bytes de memória são reservados.
- Para clientes que iniciaram de 16 de março de 2017 em diante, quando você criar um grupo de replicação ou cluster do Valkey ou Redis OSS usando o grupo de parâmetros padrão, o parâmetro de gerenciamento de memória será `reserved-memory-percent`. Nesse caso, 25% do valor `maxmemory` dos seus nós será reservado para fins não de dados.

Depois de ler sobre os dois parâmetros de gerenciamento de memória do Valkey ou Redis OSS, você pode preferir usar aquele que não é o padrão, ou com valores não padrão. Se esse for o caso, você poderá alterar para o outro parâmetro de gerenciamento de memória reservada.

Para alterar o valor desse parâmetro, você pode criar um grupo de parâmetros personalizado e modificá-lo para usar o parâmetro e o valor de gerenciamento de memória preferidos. Você pode usar o grupo de parâmetros personalizado sempre que criar um novo cluster ou grupo de replicação do Valkey ou Redis OSS. Para clusters ou grupos de replicação existentes, você pode modificá-los para usar o `parameter group` personalizado.

Para obter mais informações, consulte:

- [Especificação do parâmetro de gerenciamento de memória reservada](#)
- [Criação de um grupo de parâmetros do ElastiCache](#)

- [Modificação de um grupo de parâmetros do ElastiCache](#)
- [Modificação de um cluster do ElastiCache](#)
- [Modificação de um grupo de replicação](#)

O parâmetro reserved-memory

Antes de 16 de março de 2017, todo o gerenciamento de memória reservado do ElastiCache para Redis OSS era feito com o uso do parâmetro `reserved-memory`. O valor padrão de `reserved-memory` é 0. Este padrão não reserva memória para a sobrecarga do Valkey ou Redis OSS e permite que o Valkey ou Redis OSS consuma toda a memória de um nó com dados.

A alteração do parâmetro `reserved-memory` para que você tenha memória suficiente disponível para backups e failovers requer que você crie um `parameter group` personalizado. Neste grupo de parâmetros personalizado, você define `reserved-memory` como um valor apropriado para a versão do Valkey ou Redis OSS executada no seu cluster e o tipo de nó do cluster. Para obter mais informações, consulte [De quanta memória reservada você precisa?](#)

O parâmetro `reserved-memory` é específico do ElastiCache e não faz parte da distribuição geral do Redis OSS.

O procedimento a seguir mostra como usar `reserved-memory` para gerenciar a memória no seu cluster do Valkey ou Redis OSS.

Para reservar memória usando `reserved-memory`

1. Crie um `parameter group` personalizado especificando a família de `parameter groups` que corresponde à versão do mecanismo que você está executando, por exemplo, especificando a família de `parameter groups` `redis2.8`. Para obter mais informações, consulte [Criação de um grupo de parâmetros do ElastiCache](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis6x-m3x1 \  
  --description "Redis OSS 2.8.x for m3.xlarge node type" \  
  --cache-parameter-group-family redis6.x
```

2. Calcule quantos bytes de memória reservar para a sobrecarga do Valkey ou Redis OSS. Você pode encontrar o valor de `maxmemory` para o seu tipo de nó em [Parâmetros específicos de node-type do Redis OSS](#).

3. Modifique o parameter group personalizado para que o parâmetro `reserved-memory` seja o número de bytes que você calculou na etapa anterior. O seguinte exemplo da AWS CLI supõe que você esteja executando uma versão do Redis OSS anterior à 2.8.22 e precisa reservar metade da `maxmemory` do nó. Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory,  
  ParameterValue=7130316800"
```

Observe que você precisa de um grupo de parâmetros personalizado separado para cada tipo de nó utilizado, pois cada tipo de nó possui um valor de `maxmemory` diferente. Assim, cada tipo de nó precisa de um valor diferente para `reserved-memory`.

4. Modifique seu cluster ou grupo de replicação do Redis OSS para usar seu grupo de parâmetros personalizado.

O exemplo de CLI a seguir modifica o cluster `my-redis-cluster` para começar a usar imediatamente o parameter group `redis28-m3x1`. Para obter mais informações, consulte [Modificação de um cluster do ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

O exemplo de CLI a seguir modifica o grupo de replicação `my-redis-repl-grp` para começar a usar imediatamente o parameter group `redis28-m3x1`. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

O parâmetro reserved-memory-percent

Em 16 de março de 2017, o Amazon ElastiCache introduziu o parâmetro `reserved-memory-percent` e o disponibilizou em todas as versões do ElastiCache para Redis OSS. O objetivo de `reserved-memory-percent` é simplificar o gerenciamento da memória reservada em todos os seus clusters. Ele faz isso permitindo que você tenha um único parameter group para cada família de parameter groups (como `redis2.8`) para gerenciar a memória reservada dos seus clusters, independentemente do tipo de nó. O valor padrão para `reserved-memory-percent` é 25 (25%).

O parâmetro `reserved-memory-percent` é específico do ElastiCache e não faz parte da distribuição geral do Redis OSS.

Se o cluster estiver usando um tipo de nó da família `r6gd` e o uso de memória atingir 75%, a classificação de dados em níveis será acionada automaticamente. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

Para reservar memória usando `reserved-memory-percent`

Para usar `reserved-memory-percent` para gerenciar a memória no seu cluster do ElastiCache para Redis OSS, siga um destes procedimentos:

- Se você estiver executando o Redis OSS 2.8.22 ou posterior, atribua o grupo de parâmetros padrão ao cluster. O padrão de 25 por cento deve ser adequado. Caso contrário, siga as etapas descritas a seguir para alterar o valor.
- Se você estiver executando uma versão do Redis OSS anterior à 2.8.22, provavelmente precisará reservar mais memória do que o padrão de 25% de `reserved-memory-percent`. Para fazer isso, use o procedimento a seguir.

Para alterar o valor percentual de `reserved-memory-percent`

1. Crie um parameter group personalizado especificando a família de parameter groups que corresponde à versão do mecanismo que você está executando, por exemplo, especificando a família de parameter groups `redis2.8`. Um grupo de parâmetros personalizado é necessário, pois você não pode modificar um grupo de parâmetros padrão. Para obter mais informações, consulte [Criação de um grupo de parâmetros do ElastiCache](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --description "Redis OSS 2.8.x 50% reserved" \  
  --parameter-names reserved-memory-percent
```



```
--cache-parameter-group-family redis2.8
```

Como `reserved-memory-percent` reserva a memória como uma porcentagem da `maxmemory` de um nó, você não precisa de um `parameter group` personalizado para cada tipo de nó.

2. Modifique o `parameter group` personalizado para que `reserved-memory-percent` seja 50 (50%). Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --parameter-name-values "ParameterName=reserved-memory-percent,  
  ParameterValue=50"
```

3. Use esse grupo de parâmetros personalizado para qualquer cluster ou grupo de replicação do Redis OSS que esteja executando uma versão do Redis OSS anterior à 2.8.22.

O exemplo da CLI a seguir modifica o cluster do Redis OSS `my-redis-cluster` para começar a usar imediatamente o grupo de parâmetros `redis28-50`. Para obter mais informações, consulte [Modificação de um cluster do ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

O exemplo da CLI a seguir modifica o grupo de replicação do Redis OSS `my-redis-repl-grp` para começar a usar imediatamente o grupo de parâmetros `redis28-50`. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

Especificação do parâmetro de gerenciamento de memória reservada

Se você já era cliente ativo do ElastiCache em 16 de março de 2017, seu parâmetro de gerenciamento de memória reservada padrão é `reserved-memory` com zero (0) byte de memória reservada. Se você se tornou cliente do ElastiCache depois de 16 de março de 2017, seu parâmetro de gerenciamento de memória reservada padrão é `reserved-memory-percent`, com 25 por cento da memória do nó reservada. Isso se aplica sempre que você cria um cluster ou grupo de replicação do ElastiCache para Redis OSS. Entretanto, você pode alterar seu parâmetro de gerenciamento de memória reservada usando a AWS CLI ou a API do ElastiCache.

Os parâmetros `reserved-memory` e `reserved-memory-percent` são mutuamente exclusivos. Um grupo de parâmetros sempre terá um, mas nunca os dois. Você pode alterar o parâmetro usado por um parameter group para o gerenciamento da memória reservada, modificando o parameter group. O grupo de parâmetros deve ser personalizado, pois não é possível modificar grupos de parâmetros padrão. Para obter mais informações, consulte [Criação de um grupo de parâmetros do ElastiCache](#).

Para especificar `reserved-memory-percent`

Para usar `reserved-memory-percent` como o parâmetro de gerenciamento de memória reservada, modifique um grupo de parâmetros usando o comando `modify-cache-parameter-group`. Use o parâmetro `parameter-name-values` para especificar `reserved-memory-percent` e um valor para ele.

O seguinte exemplo de CLI modifica o parameter group personalizado `redis32-cluster-on` para que ele use `reserved-memory-percent` para gerenciar a memória reservada. Um valor deve ser atribuído a `ParameterValue` para que o grupo de parâmetros use o parâmetro `ParameterName` para gerenciamento de memória reservada. Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-cluster-on \  
  --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

Para especificar `reserved-memory`

Para usar `reserved-memory` como o parâmetro de gerenciamento de memória reservada, modifique um grupo de parâmetros usando o comando `modify-cache-parameter-group`. Use o parâmetro `parameter-name-values` para especificar `reserved-memory` e um valor para ele.

O seguinte exemplo de CLI modifica o parameter group personalizado `redis32-m3x1` para que ele use `reserved-memory` para gerenciar a memória reservada. Um valor deve ser atribuído a `ParameterValue` para que o grupo de parâmetros use o parâmetro `ParameterName` para gerenciamento de memória reservada. Como a versão do mecanismo é mais recente do que 2.8.22, definimos o valor como `3565158400`, que equivale a 25% da `cache.m3.xlarge` de um `maxmemory`. Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

```
aws elasticache modify-cache-parameter-group \
  --cache-parameter-group-name redis32-m3x1 \
  --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

Práticas recomendadas ao trabalhar com clusters baseados em nós do Valkey e Redis OSS

O uso do Multi-AZ, ter memória suficiente, redimensionar clusters e minimizar o tempo de inatividade são conceitos úteis a serem considerados ao trabalhar com clusters baseados em nós no Valkey ou Redis OSS. É recomendável analisar e seguir essas práticas recomendadas.

Tópicos

- [Minimização do tempo de inatividade com multi-AZ](#)
- [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#)
- [Redimensionamento de cluster on-line](#)
- [Minimização do tempo de inatividade durante a manutenção](#)

Minimização do tempo de inatividade com multi-AZ

Há várias instâncias em que o ElastiCache Valkey ou Redis OSS pode precisar substituir um nó primário; elas incluem certos tipos de manutenção planejada e o evento improvável de falha em um nó primário ou zona de disponibilidade.

Essa substituição resulta em algum tempo de inatividade do cluster, mas se o multi-AZ estiver habilitado, o tempo de inatividade será minimizado. A função do nó primário fará failover automaticamente para uma das réplicas de leitura. Não há necessidade de criar e provisionar um novo nó primário, porque o ElastiCache lidará com isso de forma transparente. O failover e a promoção de réplica garantem que você possa continuar a gravar no novo primário assim que a promoção estiver concluída.

Para saber mais sobre o Multi-AZ e a minimização do tempo de inatividade, consulte [Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#).

Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS

Snapshots e sincronizações no Valkey 7.2 e posterior e no Redis OSS versão 2.8.22 e posterior

O Valkey tem suporte padrão para snapshots e sincronizações. O Redis OSS 2.8.22 apresenta um processo de salvamento sem bifurcação que permite alocar mais da sua memória ao uso do seu aplicativo sem incorrer em um maior uso de permuta durante sincronizações e salvamentos. Para obter mais informações, consulte [Como a sincronização e o backup são implementados](#).

Snapshots e sincronizações do Redis OSS antes da versão 2.8.22

Quando você trabalha com o ElastiCache para Redis OSS, o Redis OSS chama um comando de gravação em segundo plano em vários casos:

- Ao criar um snapshot para um backup.
- Ao sincronizar réplicas com o primário em um grupo de replicação.
- Ao habilitar o atributo de arquivo somente de acréscimo (AOF) para o Redis OSS.
- Ao promover uma réplica para primário (o que causa uma sincronização entre o primário/réplica).

Sempre que o Redis OSS executa um processo de gravação em segundo plano, você deve ter memória disponível suficiente para acomodar a sobrecarga de processos. A falta de memória suficiente disponível faz com que o processo falhe. Por isso, é importante escolher um tipo de instância de nó que tenha memória suficiente ao criar seu cluster do Redis OSS.

Processo de gravação em segundo plano e uso de memória com o Valkey e Redis OSS

Sempre que um processo de gravação em segundo plano é chamado, o Valkey e Redis OSS bifurcam seu processo (lembre-se, esses mecanismos são single-threaded). Uma bifurcação persiste seus dados no disco em um arquivo de snapshot .rdb do Redis OSS. A outra bifurcação atende a todas as operações de leitura e gravação. Para garantir que seu snapshot seja um snapshot de ponto no tempo, todas as atualizações e adições de dados são gravadas em uma área de memória disponível separada da área de dados.

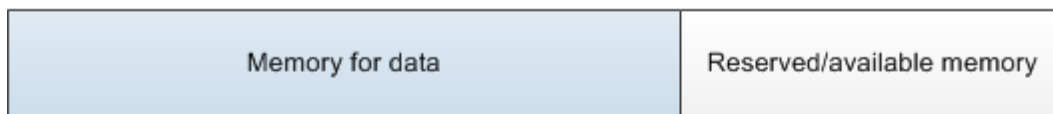
Enquanto você tiver memória suficiente disponível para registrar todas as operações de gravação durante a manutenção dos dados no disco, não haverá problemas de memória insuficiente. É

provável que você experimente problemas de memória insuficiente se uma das seguintes condições for verdadeira:

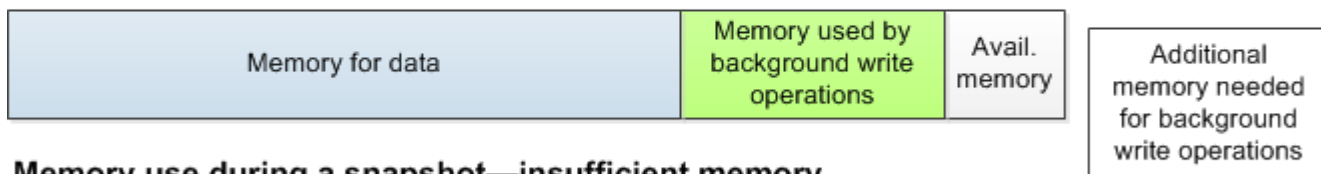
- Seu aplicativo realiza muitas operações de gravação, exigindo assim uma grande quantidade de memória disponível para aceitar os dados novos ou atualizados.
- Você tem pouca memória disponível para gravar dados novos ou atualizados.
- Você tem um grande conjunto de dados que demora muito para persistir no disco, exigindo um grande número de operações de gravação.

O diagrama a seguir ilustra o uso da memória ao executar um processo de gravação em segundo plano.

Memory use prior to a snapshot



Memory use during a snapshot—sufficient memory



Memory use during a snapshot—insufficient memory



Para obter informações sobre o impacto de fazer um backup sobre o desempenho, consulte [Impacto sobre o desempenho dos backups de clusters baseados em nós](#).

Para obter mais informações sobre como o Valkey e Redis OSS realizam snapshots, consulte <http://valkey.io>.

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#).

Evitando memória insuficiente ao executar uma gravação em segundo plano

Sempre que um processo de gravação em segundo plano, como BGSAVE ou BGREWRITEAOF, é chamado, para evitar que o processo falhe, você deve ter mais memória disponível do que será consumida por operações de gravação durante o processo. O cenário de pior caso é aquele em

que, durante a operação de gravação em segundo plano, cada registro é atualizado e alguns novos registros são adicionados ao cache. Por conta disso, recomendamos que você defina `reserved-memory-percent` como 50 (50 por cento) para versões do Redis OSS anteriores à 2.8.22 ou como 25 (25 por cento) para o Valkey e todas as versões do Redis OSS 2.8.22 e posteriores.

O valor `maxmemory` indica a memória disponível para dados e sobrecargas operacionais. Como você não pode modificar o parâmetro `reserved-memory` no `parameter group` padrão, deve criar um `parameter group` personalizado para o cluster. O valor padrão para `reserved-memory` é 0, o que permite que o Redis OSS consuma todo o `maxmemory` com dados, potencialmente deixando pouca memória para outros usos, como um processo de gravação em segundo plano. Para valores `maxmemory` por tipo de instância de nó, consulte [Parâmetros específicos de node-type do Redis OSS](#).

Você também pode usar o parâmetro `reserved-memory` para reduzir a quantidade de memória usada na caixa.

Para obter mais informações sobre parâmetros específicos do Valkey e Redis no ElastiCache, consulte [Parâmetros do Valkey e do Redis OSS](#).

Para obter informações sobre como criar e modificar `parameter groups`, consulte [Criação de um grupo de parâmetros do ElastiCache](#) e [Modificação de um grupo de parâmetros do ElastiCache](#).

Redimensionamento de cluster on-line

A refragmentação consiste na adição e remoção de fragmentos ou nós de seu cluster bem como na redistribuição de espaços importantes. Como resultado, vários itens têm impacto na operação de refragmentação, como a carga no cluster, a utilização de memória e o tamanho geral dos dados. Para obter a melhor experiência, recomendamos que você siga as práticas gerais recomendadas de cluster para distribuição padrão uniforme de workload. Além disso, recomendamos as etapas a seguir.

Antes de iniciar a refragmentação, recomendamos o seguinte:

- Teste sua aplicação: teste o comportamento da sua aplicação durante a refragmentação em um ambiente de preparação, se possível.
- Receba uma notificação prévia de problemas de escalabilidade: a refragmentação é uma operação que demanda uso intensivo de computação. Por esse motivo, recomendamos manter a utilização da CPU abaixo de 80% em instâncias de vários núcleos e abaixo de 50% em instâncias de núcleo único durante a refragmentação. Monitore as métricas do ElastiCache para Redis

OSS e inicie a refragmentação antes que a sua aplicação comece a observar problemas de escalabilidade. As métricas úteis para acompanhar são `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` e `BytesUsedForCacheItems`.

- Garanta memória livre suficiente disponível antes da redução de escala na horizontal: se você estiver reduzindo a escala na horizontal, garanta que a memória livre disponível nos fragmentos a serem retidos é, pelo menos, 1,5 vez maior do que a memória usada nos fragmentos que você planeja remover.
- Inicie a refragmentação em horários fora de pico: essa prática ajuda a reduzir a latência e o impacto de throughput no cliente durante a operação de refragmentação. Ela também ajuda a concluir a refragmentação com mais rapidez à medida que mais recursos podem ser usados na redistribuição de slots.
- Analise o comportamento de tempo limite do cliente: alguns clientes podem observar maior latência durante o redimensionamento de cluster online. Configurar sua biblioteca de cliente com um tempo limite maior pode ajudar dando tempo para o sistema se conectar mesmo em condições de carga maiores no servidor. Em alguns casos, você pode abrir um grande número de conexões com o servidor. Nesses casos, considere adicionar o recuo exponencial para uma nova conexão lógica. Fazer isso pode ajudar a evitar uma intermitência de novas conexões acessando o servidor ao mesmo tempo.
- Carregue suas funções em cada fragmento: ao aumentar a escala horizontalmente do cluster, o ElastiCache replicará automaticamente as funções carregadas em um dos nós existentes (selecionados aleatoriamente) para os novos nós. Se seu cluster tiver o Valkey 7.2 ou posterior, ou Redis OSS 7.0 ou posterior, e sua aplicação usar [Funções](#), recomendamos carregar todas as suas funções em todos os fragmentos antes de aumentar a escala horizontalmente para que seu cluster não tenha funções diferentes em fragmentos diferentes.

Após a refragmentação, observe o seguinte:

- A redução da escala horizontalmente pode ser parcialmente bem-sucedida se não houver memória suficiente disponível nos fragmentos de destino. Se isso ocorrer, analise a memória disponível e refaça a operação, se necessário. Os dados nos fragmentos de destino não serão excluídos.
- Os comandos `FLUSHALL` e `FLUSHDB` não são compatíveis em scripts Lua durante uma operação de reestilhecimento. Antes do Redis OSS 6, o comando `BRPOPLPUSH` não é compatível caso opere no slot que está sendo migrado.

Minimização do tempo de inatividade durante a manutenção

A configuração do modo de cluster tem a melhor disponibilidade durante operações gerenciadas e não gerenciadas. Recomendamos usar um cliente compatível com o modo de cluster que se conecta ao endpoint de descoberta de cluster. Para o modo de cluster desabilitado recomendamos usar o endpoint principal para todas as operações de gravação.

Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó no seu aplicativo. É por isso que, para o modo de cluster desabilitado, recomendamos que você use o endpoint do leitor para a atividade de leitura.

Se o failover automático estiver habilitado no cluster, o nó primário poderá ser alterado. Portanto, o aplicativo deve confirmar a função do nó e atualizar todos os endpoints de leitura. Fazer isso ajuda a garantir que não haja sobrecarga no nó principal. Com o failover automático desabilitado, a função do nó não é alterada. No entanto, o tempo de inatividade em operações gerenciadas ou não gerenciadas é maior em comparação ao dos clusters com failover automático habilitado.

Evite direcionar solicitações de leitura para um único nó de réplica de leitura, pois a indisponibilidade pode causar uma interrupção na leitura. Faça o fallback para a leitura pelo primário ou garanta que você tenha pelo menos duas réplicas de leitura para evitar qualquer interrupção de leitura durante a manutenção.

Estratégias de armazenamento em cache para Memcached

No tópico a seguir, você pode encontrar estratégias para preencher e manter seu cache do Memcached.

Que estratégias implementar para preencher e manter seu cache dependem de quais dados você está armazenando em cache e dos padrões de acesso a eles. Por exemplo, você provavelmente não quer usar a mesma estratégia para um placar dos 10 primeiros colocados em um site de jogos e em notícias atuais. No restante desta seção, discutiremos estratégias comuns de manutenção de cache, suas vantagens e suas desvantagens.

Tópicos

- [Réplicas de leitura](#)
- [Carregamento lento](#)
- [Gravação simultânea](#)
- [Adicionar o TTL](#)
- [Tópicos relacionados](#)

Réplicas de leitura

Muitas vezes, você pode melhorar significativamente o desempenho dos caches com tecnologia sem servidor do ElastiCache criando réplicas e lendo-as em vez do nó de cache primário. Para obter mais informações, consulte [Práticas recomendadas para usar réplicas de leitura](#).

Carregamento lento

Como o nome indica, o carregamento lento é uma estratégia de cache que carrega dados no cache apenas quando necessário. Ele funciona como descrito a seguir.

O Amazon ElastiCache é um repositório de chaves-valores na memória entre a aplicação e o armazenamento de dados (banco de dados) que ele acessa. Sempre que a sua aplicação solicita dados, ela primeiro faz a solicitação ao cache do ElastiCache. Se os dados existirem no cache e forem atuais, o ElastiCache os retornará à sua aplicação. Se os dados não existirem no cache ou tiverem expirado, sua aplicação solicitará os dados do seu armazenamento de dados. Seu armazenamento de dados retorna os dados para sua aplicação. Sua aplicação, em seguida, grava os dados recebidos do armazenamento no cache. Dessa forma, eles podem ser recuperados mais rapidamente na próxima vez que forem solicitados.

Um acerto de cache ocorre quando os dados estão no cache e não estão expirados:

1. A aplicação solicita dados do cache.
2. O cache retorna os dados à aplicação.

Uma perda de cache ocorre quando os dados não estão no cache ou estão expirados:

1. Sua aplicação solicita dados do cache.
2. O cache não possui os dados solicitados e, portanto, retorna `null`.
3. Sua aplicação solicita e recebe os dados do banco de dados.
4. A aplicação atualiza o cache com os novos dados.

Vantagens e desvantagens do carregamento lento

As vantagens do carregamento lento são as seguintes:

- Apenas dados solicitados são armazenados em cache.

Como a maioria dos dados nunca é solicitada, o carregamento lento evita o preenchimento do cache com dados que não são solicitados.

- Falhas de nó não são fatais para sua aplicação.

Quando um nó falha e é substituído por um novo nó vazio, sua aplicação continua a funcionar, embora com latência aumentada. À medida que as solicitações são feitas para o novo nó, cada perda de cache resulta em uma consulta do banco de dados. Ao mesmo tempo, a cópia de dados é adicionada ao cache para que as solicitações subsequentes sejam recuperadas do cache.

As desvantagens do carregamento lento são as seguintes:

- Existe uma penalidade de ausência no cache. A perda de cache resulta em 3 viagens,
 1. Solicitação inicial de dados do cache
 2. Consulta do banco de dados para os dados
 3. Gravação de dados no cache

Essas perdas podem causar um atraso observável na obtenção dos dados para a aplicação.

- Dados obsoletos.

Se os dados forem gravados no cache somente quando houver uma perda de cache, os dados no cache poderão ficar obsoletos. Esse resultado ocorre porque não há atualizações para o cache quando os dados são alterados no banco de dados. Para resolver esse problema, você pode usar as estratégias [Gravação simultânea](#) e [Adicionar o TTL](#).

Exemplo de pseudocódigo de carregamento lento

O código a seguir é um exemplo de pseudocódigo de lógica de carregamento lento.

```
// *****  
// 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
```

Para este exemplo, o código da aplicação que obtém os dados é o seguinte.

```
customer_record = get_customer(12345)
```

Gravação simultânea

A estratégia de gravação simultânea adiciona dados ou atualiza dados no cache sempre que eles são gravados no banco de dados.

Vantagens e desvantagens da gravação simultânea

As vantagens da gravação simultânea são as seguintes:

- Os dados no cache nunca são obsoletos.

Como os dados no cache são atualizados sempre que são escritos no banco de dados, eles sempre estão atualizados.

- Penalidade de gravação versus penalidade de leitura.

Cada gravação envolve duas viagens:

1. Uma gravação no cache
2. Uma gravação no banco de dados

Que acrescenta latência ao processo. Dito isto, os usuários finais geralmente são mais tolerantes à latência ao atualizarem dados do que ao recuperar dados. Existe um sentido inerente de que as atualizações são mais trabalhosas e, portanto, demoram mais.

As desvantagens da gravação simultânea são as seguintes:

- Dados ausentes.

Se você cria um novo nó, seja devido a uma falha de nó ou ao aumento de escala na horizontal, haverá dados ausentes. Esses dados continuam ausentes até que sejam adicionados ou atualizados no banco de dados. Você pode minimizar isso implementando [carregamento lento](#) com gravação simultânea.

- Rotatividade do cache.

A maioria dos dados nunca é lida, o que é um desperdício de recursos. Ao [adicionar um valor de tempo de vida \(TTL\)](#), você pode minimizar o espaço desperdiçado.

Exemplo de pseudocódigo de gravação simultânea

O código a seguir é um exemplo de pseudocódigo da lógica de gravação simultânea.

```
// *****  
// 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
```

Para este exemplo, o código da aplicação que obtém os dados é o seguinte.

```
save_customer(12345, {"address": "123 Main"})
```

Adicionar o TTL

O carregamento lento permite dados obsoletos, mas não falha com nós vazios. A gravação simultânea garante que os dados sempre sejam atuais, mas pode falhar com nós vazios e pode preencher o cache com dados supérfluos. Ao adicionar um valor de tempo de vida (TTL) a cada gravação, você pode ter as vantagens de cada estratégia. Ao mesmo tempo, você pode em grande parte evitar desordenar o cache com dados adicionais.

O Time to live (TTL) (Tempo de vida) é um valor inteiro que especifica o número de segundos até a chave expirar. O Valkey ou Redis OSS pode especificar segundos ou milissegundos para este valor. O Memcached especifica este valor em segundos. Quando uma aplicação tenta ler uma chave expirada, isso é tratado como se a chave não fosse encontrada. O banco de dados é consultado para a chave e o cache é atualizado. Essa abordagem não garante que um valor não esteja obsoleto. Contudo, isso impede que os dados fiquem obsoletos demais e exige que os valores no cache sejam ocasionalmente atualizados a partir do banco de dados.

Para obter mais informações, consulte [os comandos Valkey ou Redis OSS](#) ou os [comandos Memcached set](#).

Exemplos de pseudocódigo de TTL

O código a seguir é um exemplo de pseudocódigo da lógica de gravação simultânea com TTL.

```
// *****
// 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
```

O código a seguir é um exemplo de pseudocódigo de lógica de carregamento lento com TTL.

```
// *****
// 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
```

Para este exemplo, o código da aplicação que obtém os dados é o seguinte.

```
save_customer(12345, {"address": "123 Main"})
```

```
customer_record = get_customer(12345)
```

Tópicos relacionados

- [Armazenamento de dados na memória](#)
- [Escolha de um mecanismo e versão](#)

- [Dimensionamento ElastiCache](#)

Gerenciando seu cluster baseado em nós no ElastiCache

ElastiCache oferece duas opções de implantação, caches sem servidor e clusters baseados em nós. Cada um tem suas próprias capacidades e requisitos.

Esta seção contém tópicos para ajudar você a gerenciar clusters baseados em nós.

Note

Esses tópicos não se aplicam ao ElastiCache Serverless.

Tópicos

- [Auto Scaling de clusters Valkey e Redis OSS](#)
- [Modificar o modo de cluster](#)
- [Replicação entre AWS regiões usando datastores globais](#)
- [Alta disponibilidade com o uso de grupos de replicação](#)
- [Gerenciamento de manutenção de cluster do ElastiCache](#)
- [Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache](#)

Auto Scaling de clusters Valkey e Redis OSS

Pré-requisitos

ElastiCache O Auto Scaling está limitado ao seguinte:

- Clusters Valkey ou Redis OSS (modo cluster habilitado) que executem o Valkey 7.2 em diante ou que executem o Redis OSS 6.0 em diante
- Clusters em camadas de dados (modo cluster habilitado) que executem o Valkey 7.2 em diante ou que executem o Redis OSS 7.0.7 em diante
- Tamanhos de instância - Grande XLarge, 2 XLarge
- Famílias de tipo de instância: R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn

- O Auto Scaling in não ElastiCache é compatível com clusters executados em datastores globais, Outposts ou Locais Zones.

Gerenciando a capacidade automaticamente com o ElastiCache Auto Scaling com Valkey ou Redis OSS

ElastiCache o escalonamento automático com Valkey ou Redis OSS é a capacidade de aumentar ou diminuir automaticamente os fragmentos ou réplicas desejados em seu serviço. ElastiCache ElastiCache aproveita o serviço Application Auto Scaling para fornecer essa funcionalidade. Para obter mais informações, consulte [Application Auto Scaling](#). Para usar o escalonamento automático, você define e aplica uma política de escalabilidade que usa CloudWatch métricas e valores-alvo que você atribui. ElastiCache o auto scaling usa a política para aumentar ou diminuir o número de instâncias em resposta às cargas de trabalho reais.

Você pode usar o Console de gerenciamento da AWS para aplicar uma política de escalabilidade com base em uma métrica predefinida. Uma `predefined metric` é definida em uma enumeração, para que você possa especificá-la por nome no código ou usá-la no Console de gerenciamento da AWS. As métricas personalizadas não estão disponíveis para seleção ao usar o Console de gerenciamento da AWS. Como alternativa, você pode usar a API Application Auto Scaling AWS CLI ou a Application Auto Scaling para aplicar uma política de escalabilidade com base em uma métrica predefinida ou personalizada.

ElastiCache para Valkey e Redis, o OSS suporta escalabilidade para as seguintes dimensões:

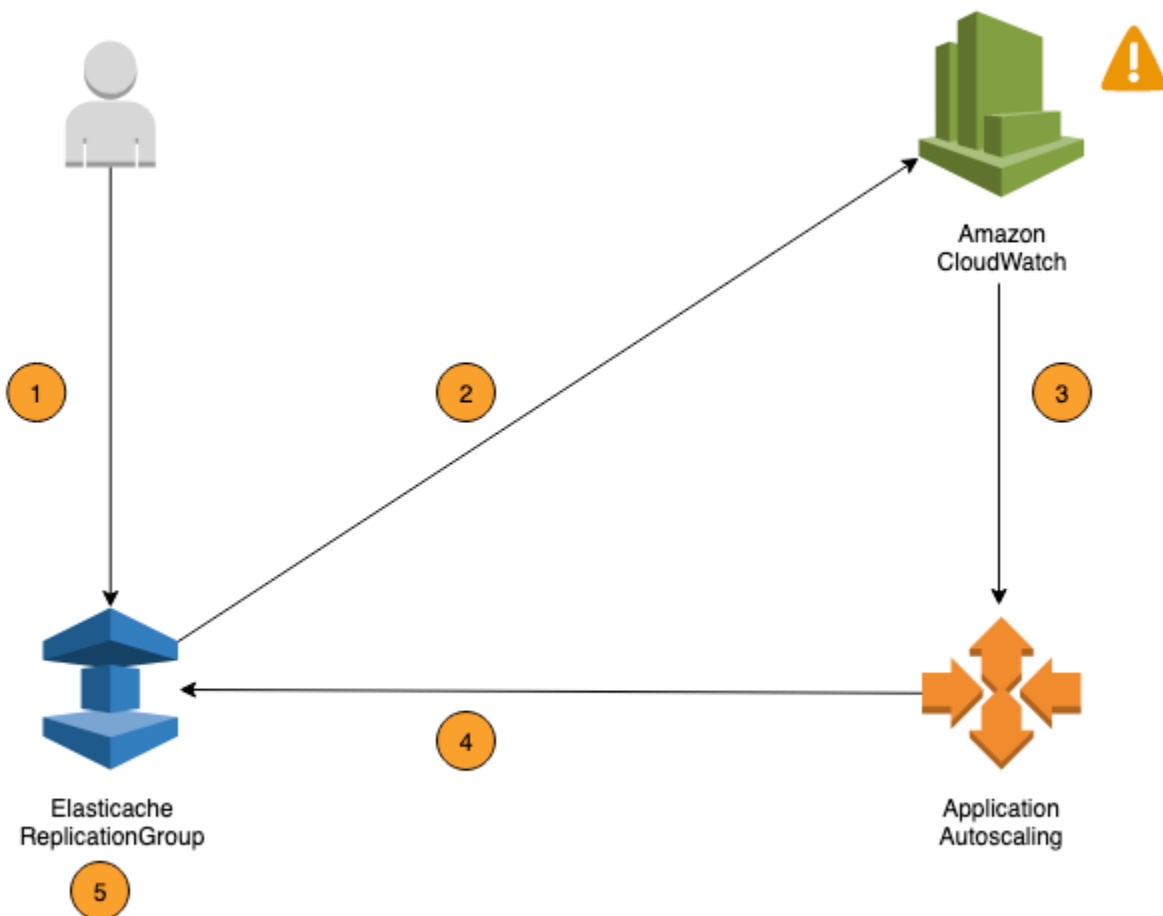
- Fragmentos: adicionar/remover fragmentos automaticamente no cluster semelhantes à refragmentação manual online. Nesse caso, o escalonamento ElastiCache automático aciona o escalonamento em seu nome.
- Réplicas — add/remove replicas in the cluster similar to manual Increase/Decrease replica operations. ElastiCache auto scaling for Valkey and Redis OSS adds/removes replica automaticamente de maneira uniforme em todos os fragmentos do cluster.

ElastiCache para Valkey e Redis, o OSS oferece suporte aos seguintes tipos de políticas de escalabilidade automática:

- [Políticas de escalabilidade de rastreamento de destino](#)— Aumente ou diminua o número de shards/réplicas execuções do seu serviço com base em um valor alvo para uma métrica

específica. Isso é semelhante à forma como o termostato mantém a temperatura da casa. Você seleciona a temperatura, e o termostato faz o resto.

- [Escalabilidade programada para seu aplicativo.](#) — ElastiCache para Valkey e Redis, o escalonamento automático do OSS pode aumentar ou diminuir o número shards/replicas de execuções do seu serviço com base na data e hora.



As etapas a seguir resumem o ElastiCache processo de escalonamento automático do Valkey e do Redis OSS, conforme mostrado no diagrama anterior:

1. Você cria uma política de escalabilidade ElastiCache automática para seu grupo de replicação.
2. ElastiCache o auto scaling cria um par de CloudWatch alarmes em seu nome. Cada par representa seus limites superiores e inferiores para métricas. Esses CloudWatch alarmes são

acionados quando a utilização real do cluster se desvia da utilização desejada por um longo período de tempo. Agora, é possível visualizar os alarmes no console.

3. Se o valor da métrica configurada exceder sua meta de utilização (ou ficar abaixo da meta) por um período de tempo específico, CloudWatch acionará um alarme que invoca o escalonamento automático para avaliar sua política de escalabilidade.
4. ElastiCache o auto scaling emite uma solicitação de modificação para ajustar a capacidade do cluster.
5. ElastiCache processa a solicitação de modificação, aumentando (ou diminuindo) dinamicamente a Shards/Replicas capacidade do cluster para que ela se aproxime da utilização desejada.

Para entender como o ElastiCache Auto Scaling funciona, suponha que você tenha um cluster chamado `UsersCluster`. Ao monitorar as CloudWatch métricas `UsersCluster`, você determina o máximo de fragmentos que o cluster exige quando o tráfego está no pico e o mínimo de fragmentos quando o tráfego está no ponto mais baixo. Você também decide um valor-alvo para a utilização da CPU no `UsersCluster` cluster. ElastiCache o auto scaling usa seu algoritmo de rastreamento de metas para garantir que os fragmentos provisionados de `UsersCluster` sejam ajustados conforme necessário para que a utilização permaneça no valor alvo ou próximo dele.

Note

O escalonamento pode levar um tempo perceptível e exigirá recursos extras do cluster para que os fragmentos se rebalanceiem. ElastiCache O Auto Scaling modifica as configurações de recursos somente quando a carga de trabalho real permanece elevada (ou deprimida) por um período sustentado de vários minutos. O algoritmo de monitoramento do objetivo do ajuste de escala automático procura manter a utilização pretendida no valor escolhido ou próximo a ele em longo prazo.

Políticas de Auto Scaling

Uma política de escalabilidade tem os seguintes componentes:

- Uma métrica de destino: a métrica do CloudWatch que o ajuste de escala automático do ElastiCache para Valkey e Redis OSS usa para determinar o momento e a quantidade a escalar.
- Capacidade mínima e máxima: o número mínimo e máximo de fragmentos a ser usado para a escalabilidade.

⚠ Important

Ao criar a política de Auto Scaling, se a capacidade atual for maior do que a capacidade máxima configurada, reduzimos a escala na horizontal até MaxCapacity durante a criação da política. Da mesma forma, se a capacidade atual for menor do que a capacidade mínima configurada, aumentamos a escala na horizontal até MinCapacity.

- Um período de desaquecimento: a quantidade de tempo, em segundos, que outra ação de aumento ou redução de escala na horizontal deve aguardar antes de ser iniciada, após a conclusão de uma ação de expansão ou redução.
- Uma função vinculada a serviços: uma função do Identity and Access Management (IAM) da AWS vinculada a um serviço AWS específico. Uma função vinculada ao serviço inclui todas as permissões que o serviço exige para chamar todos os outros serviços da AWS em seu nome. O ajuste de escala automático do ElastiCache gera automaticamente essa função, `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`, para você.
- Habilitar ou desabilitar atividades de redução de escala na horizontal: capacidade de habilitar ou desabilitar atividades de redução de escala na horizontal para uma política.

Tópicos

- [Métrica de destino para o Auto Scaling](#)
- [Capacidades mínima e máxima](#)
- [Período de desaquecimento](#)
- [Habilitar ou desabilitar atividades de redução](#)

Métrica de destino para o Auto Scaling

Neste tipo de política, uma métrica predefinida ou personalizada e um valor de destino dessa métrica são especificados na configuração de uma política de escalabilidade de rastreamento de destino. O ajuste de escala automático do ElastiCache para Valkey e Redis OSS cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove fragmentos/réplicas conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a workloads variáveis.

Essas políticas também minimizam flutuações rápidas no número de fragmentos/réplicas disponíveis para seu cluster.

Por exemplo, considere uma política de escalabilidade que use a métrica predefinida de média `ElastiCachePrimaryEngineCPUUtilization`. Essa política pode manter a utilização da CPU a um percentual de utilização específico, por exemplo, 70 por cento, ou próximo a isso.

Note

Para cada cluster, você pode criar somente uma política de Auto Scaling para cada métrica de destino.

Capacidades mínima e máxima

Fragmentos

Você pode especificar o número máximo de fragmentos que podem ser escalados pelo ajuste de escala automático do ElastiCache para Valkey e Redis OSS. Esse valor deve ser menor que ou igual a 250, com um mínimo de 1. Você também pode especificar o número mínimo de fragmentos a serem gerenciados pelo ajuste de escala automático. O valor mínimo é 1, e deve ser menor ou igual ao valor especificado para o número máximo de fragmentos de 250.

Réplicas

Você pode especificar o número máximo de réplicas a serem gerenciadas pelo ajuste de escala automático do ElastiCache para Valkey e Redis OSS. Esse valor deve ser igual ou menor que 5. Você também pode especificar o número mínimo de réplicas a serem gerenciadas pelo ajuste de escala automático. O valor mínimo é 1, e deve ser menor ou igual ao valor especificado para o número máximo de réplicas de 5.

Para determinar o número mínimo e máximo de fragmentos/réplicas que você precisa para um tráfego típico, teste a configuração do Auto Scaling com a taxa esperada de tráfego para o modelo.

Note

As políticas de escalabilidade automática do ElastiCache aumentam a capacidade do cluster até atingir o tamanho máximo definido ou até que os limites do serviço sejam aplicados. Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

 Important

Redução de escala na horizontal quando não há tráfego Se o tráfego de uma variante atingir zero, o ElastiCache automaticamente reduz a escala na horizontal para o número mínimo de instâncias especificado.

Período de desaquecimento

Você pode ajustar a responsividade das políticas de escalabilidade com monitoramento do objetivo, adicionando períodos de desaquecimento que afetam a escalabilidade de seu cluster. Um desaquecimento bloqueia solicitações subsequentes de redução ou expansão até o período expirar. Isso atrasa as exclusões de fragmentos/réplicas no cluster do ElastiCache para Valkey e Redis OSS para solicitações de redução de escala horizontal e a criação de fragmentos/réplicas para solicitações de aumento de escala horizontal. Você pode especificar os seguintes desaquecimentos:

- A atividade de redução de escala horizontal diminui o número de fragmentos/réplicas em seu cluster. Um desaquecimento de redução especifica a quantidade de tempo, em segundos, após a conclusão de uma ação de redução antes que quaisquer outras atividades de redução possam iniciar.
- A atividade de aumento de escala horizontal aumenta o número de fragmentos/réplicas em seu cluster. Um desaquecimento de expansão especifica a quantidade de tempo, em segundos, após a conclusão de uma ação de expansão antes que quaisquer outras atividades de expansão possam iniciar.

Quando o período de desaquecimento de redução ou aumento de escala na horizontal não é especificado, o padrão para o aumento é de 600 segundos, e para a redução, 900 segundos.

Habilitar ou desabilitar atividades de redução

Você pode habilitar ou desabilitar atividades de redução para as políticas. Habilitar atividades de redução de escala na horizontal permite que as políticas de escalabilidade excluam fragmentos/réplicas. Quando as atividades de redução são habilitadas, o desaquecimento de redução na política de escalabilidade aplica-se a atividades de redução. Desabilitar atividades de redução de escala na horizontal evita que as políticas de escalabilidade excluam fragmentos/réplicas.

 Note

As atividades de aumento de escala na horizontal são habilitadas para que a política de escalabilidade possa criar fragmentos ou réplicas do ElastiCache, conforme o necessário.

Permissões do IAM necessárias para o Auto Scaling

ElastiCache para Valkey e Redis, o OSS Auto Scaling é possível graças a uma combinação do ElastiCache,, e do Application CloudWatch Auto Scaling. APIs Os clusters são criados e atualizados com ElastiCache, os alarmes são criados com CloudWatch e as políticas de escalabilidade são criadas com o Application Auto Scaling. Além das permissões padrão do IAM para criar e atualizar clusters, o usuário do IAM que acessa as configurações do ElastiCache Auto Scaling deve ter as permissões apropriadas para os serviços que oferecem suporte ao escalonamento dinâmico. Nessa política mais recente, adicionamos suporte à escalabilidade vertical do Memcached, com a ação `elasticache:ModifyCacheCluster`. Os usuários do IAM precisam ter as permissões para usar as ações exibidas na política de exemplo a seguir:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:*",
        "elasticache:DescribeReplicationGroups",
        "elasticache:ModifyReplicationGroupShardConfiguration",
        "elasticache:IncreaseReplicaCount",
        "elasticache:DecreaseReplicaCount",
        "elasticache:DescribeCacheClusters",
        "elasticache:DescribeCacheParameters",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",

```

```

        "cloudwatch:DisableAlarmActions",
        "cloudwatch:EnableAlarmActions",
        "iam:CreateServiceLinkedRole",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Get*",
        "sns:List*"
    ],
    "Resource": "arn:aws:iam::123456789012:role/autoscaling-roles-for-
cluster"
    }
    ]
}

```

Perfil vinculado a serviço

O serviço de escalonamento automático OSS ElastiCache para Valkey e Redis também precisa de permissão para descrever seus clusters CloudWatch e alarmes, além de permissões para modificar ElastiCache sua capacidade alvo em seu nome. Se você ativar o Auto Scaling em seu cluster, ele criará uma função vinculada ao serviço chamada `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`. Essa função vinculada ao serviço concede permissão de escalonamento ElastiCache automático para descrever os alarmes de suas políticas, monitorar a capacidade atual da frota e modificar a capacidade da frota. A função vinculada ao serviço é a função padrão para o escalonamento ElastiCache automático. Para obter mais informações, consulte [Funções vinculadas a serviços ElastiCache para o escalonamento automático do Redis OSS no Guia do usuário do Application Auto Scaling](#).

Práticas recomendadas de escalabilidade automática

Antes de se registrar no Auto Scaling, recomendamos o seguinte:

1. Use apenas uma métrica de monitoramento: identifique se o cluster tem workloads com uso intenso da CPU ou de dados e use uma métrica predefinida correspondente para definir a política de ajuste de escala.
 - CPU do mecanismo: `ElastiCachePrimaryEngineCPUUtilization` (dimensão fragmentada) ou `ElastiCacheReplicaEngineCPUUtilization` (dimensão da réplica)
 - Uso do banco de dados: `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` essa política de ajuste de escala funciona melhor com `maxmemory-policy` definido como `noeviction` no cluster.

Recomendamos que você evite várias políticas por dimensão no cluster. ElastiCache para Valkey e Redis OSS, o Auto Scaling expandirá a meta escalável se alguma política de rastreamento de alvos estiver pronta para ser expandida, mas será ampliada somente se todas as políticas de rastreamento de alvos (com a parte de expansão ativada) estiverem prontas para serem ampliadas. Se várias políticas instruírem o destino escalável a aumentar ou reduzir a escala na horizontal ao mesmo tempo, ele escalará com base na política que forneça a maior capacidade tanto para reduzir quanto para aumentar a escala na horizontal.

2. Métricas personalizadas para monitoramento do objetivo: seja cauteloso ao usar métricas personalizadas para o monitoramento do objetivo, pois o ajuste de escala automático é mais adequado para aumentar/reduzir a escala horizontalmente de modo proporcional às alterações nas métricas escolhidas para a política. Se essas métricas não forem alteradas proporcionalmente às ações de ajuste de escala usadas para a criação de políticas, elas poderão aumentar ou reduzir a escala horizontalmente das ações de forma contínua, o que pode afetar a disponibilidade ou o custo.

Para clusters de armazenamento de dados em camadas (tipos de instância da família r6gd), evite usar métricas baseadas em memória para ajuste de escala.

3. Escalabilidade programada — Se você identificar que sua carga de trabalho é determinística (atinge high/low em um horário específico), recomendamos usar a escalabilidade programada e configurar sua capacidade alvo de acordo com a necessidade. O monitoramento do objetivo é mais adequado para workloads não determinísticas e para o cluster operar na métrica de destino necessária, aumentando a escala horizontalmente quando você precisar de mais recursos e reduzindo a escala horizontalmente quando precisar de menos recursos.
4. Desative o escalonamento — O escalonamento automático no Target Tracking é mais adequado para clusters com cargas increase/decrease de trabalho graduais, pois as métricas podem acionar oscilações consecutivas de spikes/dip escala/aumento. Para evitar tais oscilações, é possível começar com a opção de reduzir a escala horizontalmente desabilitada e, mais tarde, você pode reduzir a escala horizontalmente manualmente de acordo com sua necessidade.
5. Teste seu aplicativo — Recomendamos que você teste seu aplicativo com suas Min/Max cargas de trabalho estimadas para determinar o mínimo e o máximo absolutos shards/replicas necessários para o cluster enquanto cria políticas de escalabilidade para evitar problemas de disponibilidade. A autoescalabilidade pode aumentar a escala horizontalmente até o máximo e reduzir a escala horizontalmente até o mínimo limite configurado para o destino.
6. Definindo o valor-alvo — Você pode analisar CloudWatch as métricas correspondentes para a utilização do cluster em um período de quatro semanas para determinar o limite do valor-alvo. Se

you still don't have certainty of which value to choose, we recommend starting with the minimum of the predefined compatible metric.

7. AutoScaling on Target Tracking is more appropriate for clusters with uniform distribution of workloads across all shards/replicas dimensions. Having non-uniform distribution can lead to:

- Dimensionamento quando não é necessário devido à carga de trabalho spike/dip em alguns fragmentos/réplicas quentes.
- Não escalar quando necessário devido à média geral perto do destino, mesmo tendo fragmentos/réplicas quentes.

Note

When you scale your cluster, ElastiCache will automatically replicate the functions loaded on one of the existing nodes (selected randomly) to the new node(s). If your cluster has Valkey or Redis OSS 7.0 or later and your application uses [Functions](#), we recommend loading all your functions on all shards before increasing the scale horizontally so that your cluster does not have different functions on different shards.

After you register AutoScaling, observe the following:

- There are limitations in configurations compatible with auto-scalability. Therefore, we recommend that you do not change the configuration of a registered replication group for automatic scalability. See the examples below:
- Manual modification of the instance type for unsupported types.
- Association of the replication group to a global datastore.
- Alteração do parâmetro ReservedMemoryPercent.
- increasing/decreasing shards/replicas beyond the Min/MaxCapacidade configurada manualmente durante a criação da política.

Uso do Auto Scaling com fragmentos

With the Auto Scaling of ElastiCache, you can use programmed and monitoring policies with your Valkey or Redis OSS mechanism.

O tópico seguinte fornece detalhes sobre o monitoramento do objetivo e as políticas programadas e como aplicá-las usando o Console de gerenciamento da AWS, a AWS CLI e APIs.

Tópicos

- [Políticas de escalabilidade de rastreamento de destino](#)
- [Adicionar uma política de escalabilidade](#)
- [Registro de um destino escalável](#)
- [Definir uma política de escalabilidade](#)
- [Desabilitar a atividade de redução](#)
- [Aplicar uma política de escalabilidade](#)
- [Editar uma política de escalabilidade](#)
- [Excluir uma política de escalabilidade](#)
- [Use CloudFormation para políticas do Auto Scaling](#)
- [Escalabilidade programada](#)

Políticas de escalabilidade de rastreamento de destino

Com as políticas de dimensionamento com monitoramento do objetivo, você seleciona uma métrica e define um valor pretendido. O Auto Scaling do ElastiCache para Valkey e Redis OSS cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove fragmentos conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a um padrão de carga de flutuação, e minimiza as flutuações rápidas na capacidade da frota.

Por exemplo, considere uma política de escalabilidade que use a métrica predefinida de média `ElastiCachePrimaryEngineCPUUtilization` com valor de destino configurado. Essa política pode manter a utilização da CPU em, ou próxima do valor de destino especificado.

Métricas predefinidas

Uma métrica predefinida é uma estrutura que se refere a um nome, dimensão e estatística (average) específicos de uma determinada métrica do CloudWatch. Sua política de ajuste de escala automático define as seguintes métricas predefinidas para seu cluster:

Nome da métrica predefinida	Nome da métrica do CloudWatch	Dimensão da métrica do CloudWatch.	Tipos de instância inelegíveis
ElastiCachePrimaryEngineCPUUtilization	EngineCPUUtilization	ReplicationGroupId, Função = primária	Nenhum
ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage	DatabaseCapacityUsageCountedForEvictPercentage	Métricas do grupo de replicação do Valkey ou Redis OSS	Nenhum
ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage	DatabaseMemoryUsageCountedForEvictPercentage	Métricas do grupo de replicação do Valkey ou Redis OSS	R6gd

Os tipos de instância em camadas de dados não podem usar `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, pois esses tipos de instância armazenam dados na memória e no SSD. O caso de uso previsto para instâncias em camadas de dados é usar 100% de memória e preencher o SSD conforme necessário.

Critérios do Auto Scaling para fragmentos

Quando o serviço detectar que sua métrica predefinida é igual ou maior que a configuração de destino, ele aumentará automaticamente a capacidade dos fragmentos. O ElastiCache para Valkey e Redis OSS aumenta a escala na horizontal dos fragmentos do seu cluster em uma contagem igual à maior de dois números: variação percentual do objetivo e 20% dos fragmentos atuais. Para reduzir a

escala horizontalmente, o ElastiCache não fará a redução da escala na horizontal automaticamente a menos que o valor geral da métrica seja inferior a 75% do seu objetivo definido.

Para um exemplo de aumento de escala na horizontal, se você tiver 50 fragmentos e

- se o seu objetivo violar em 30%, o ElastiCache aumentará a escala na horizontal em 30%, o que resultará em 65 fragmentos por cluster.
- Se o seu objetivo violar em 10%, o ElastiCache aumentará a escala na horizontal em um mínimo padrão de 20%, o que resultará em 60 fragmentos por cluster.

Para um exemplo de redução de escala na horizontal, se você selecionou um valor de objetivo de 60%, o ElastiCache não reduzirá a escala na horizontal automaticamente até que a métrica seja menor ou igual a 45% (25% abaixo do objetivo de 60%).

Considerações sobre o Auto Scaling

Lembre-se das seguintes considerações:

- Uma política de escalabilidade de rastreamento de destino pressupõe que ela deve aumentar a escalabilidade quando a métrica especificada estiver acima do valor de destino. Você não pode usar uma política de escalabilidade de rastreamento de destino para expandir quando a métrica especificada estiver abaixo do valor de destino. O ElastiCache para Valkey e Redis OSS aumenta a escala na horizontal de fragmentos em um desvio mínimo de 20% do objetivo dos fragmentos existentes no cluster.
- Uma política de escalabilidade de rastreamento de destino não escala quando a métrica especificada tem dados insuficientes. Ela não reduz a escala horizontalmente, porque não interpreta dados insuficientes como baixa utilização.
- É possível ver lacunas entre o valor de destino e os pontos de dados de métrica reais. Isso ocorre porque o Auto Scaling do ElastiCache funciona de maneira segura por arredondamento para cima ou para baixo, quando ele determina a capacidade a ser adicionada ou removida. Isso evita que ele adicione capacidade insuficiente ou remova muita capacidade.
- Para garantir a disponibilidade da aplicação, o serviço aumenta a escala na horizontal proporcionalmente à métrica o mais rápido possível, mas é reduz a escala na horizontal de forma mais conservadora.
- Você pode ter várias políticas de escalabilidade com monitoramento do objetivo para um cluster do ElastiCache para Valkey e Redis OSS, desde que cada uma delas use uma métrica diferente. A intenção do ajuste de escala automático do ElastiCache é sempre priorizar a disponibilidade.

Portanto, seu comportamento será diferente, se as políticas de monitoramento do objetivo estiverem prontas ou não para aumentar ou reduzir a escala horizontalmente. Ele vai aumentar o serviço se qualquer uma das políticas de monitoramento do objetivo estiverem prontas para aumentar, mas vai reduzir somente se todas as políticas de monitoramento do objetivo (com a parte de redução habilitada) estiverem prontas para reduzir.

- Não edite nem exclua os alarmes do CloudWatch que o ajuste de escala automático do ElastiCache gerencia para uma política de escalabilidade com monitoramento do destino. O Auto Scaling do ElastiCache exclui os alarmes automaticamente quando você exclui a política de escalabilidade.
- O Auto Scaling do ElastiCache não impede a modificação manual de fragmentos do cluster. Esses ajustes manuais não afetam alarmes existentes do CloudWatch que estejam conectados à política de escalabilidade, mas podem afetar métricas que possam acionar esses alarmes do CloudWatch.
- Esses alarmes do CloudWatch gerenciados pelo Auto Scaling são definidos na métrica do AVG em todos os fragmentos do cluster. Assim, ter fragmentos quentes pode resultar em qualquer cenário de:
 - escalabilidade quando não for necessário devido à carga em alguns fragmentos quentes que acionam um alarme do CloudWatch
 - não escalar quando necessário devido ao AVG agregado em todos os fragmentos que afetam o alarme não violarem.
- Os limites padrão do ElastiCache em nós por cluster ainda se aplicam. Então, ao optar pelo Auto Scaling, se você espera que os nós máximos sejam mais do que o limite padrão, solicite um aumento de limite em [Limites de serviço da AWS](#) e escolha o tipo de limite Nós por cluster por tipo de instância.
- Certifique-se de que você tenha ENIs (interfaces de rede elásticas) suficientes disponíveis em sua VPC, que são necessárias durante o aumento de escala na horizontal. Para obter mais informações, consulte [Interfaces de rede elástica](#).
- Se não houver capacidade suficiente disponível do EC2, o Auto Scaling do ElastiCache não ajustará a escala e atrasará até que a capacidade esteja disponível.
- O Auto Scaling do ElastiCache para Redis OSS, durante a redução de escala na horizontal, não removerá fragmentos com slots com um tamanho de item maior que 256 MB pós-serialização.
- Durante a redução de escala na horizontal, ele não removerá fragmentos se houver memória insuficiente disponível na configuração de fragmento resultante.

Adicionar uma política de escalabilidade

Você pode adicionar uma política de escalabilidade usando o Console de gerenciamento da AWS.

Para adicionar uma política do Auto Scaling a um cluster do ElastiCache para Valkey e Redis OSS

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Selecione o cluster ao qual você deseja adicionar uma política (escolha o botão à esquerda do nome do cluster).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Escolha add dynamic scaling (adicionar escalabilidade dinâmica).
6. Em Policy Name (Nome da política), insira um nome para a política.
7. Em Scalable Dimension (Dimensão escalável), escolha shards (fragmentos).
8. Quanto à métrica de destino, escolha uma das seguintes opções:
 - Primary CPU Utilization (Utilização da CPU primária) para criar uma política com base na utilização média da CPU.
 - Memory (Memória) para criar uma política com base na memória média do banco de dados.
 - Capacidade de criar uma política com base no uso médio da capacidade do banco de dados. A métrica de capacidade inclui a utilização de memória e SSD para instâncias em camadas de dados e a utilização da memória para todos os outros tipos de instância.
9. Para o valor alvo, escolha um valor maior ou igual a 35 e menor ou igual a 70. O ajuste de escala automático manterá esse valor para a métrica de destino selecionada em seus fragmentos do ElastiCache:
 - Utilização primária da CPU: mantém o valor alvo da métrica EngineCPUUtilization nos nós primários.
 - Memória: mantém o valor alvo da métrica DatabaseMemoryUsageCountedForEvictPercentage
 - Capacidade: mantém o valor alvo da métrica DatabaseCapacityUsageCountedForEvictPercentage

Os fragmentos do cluster serão adicionados ou removidos para manter a métrica próxima ao valor especificado.

10. (Opcional) Não há suporte do console para os períodos de desaquecimento de redução ou aumento de escala na horizontal. Use a AWS CLI para modificar os valores de desaquecimento.
11. Em Capacidade mínima, insira o número mínimo de fragmentos que a política do Auto Scaling do ElastiCache é solicitada a manter.
12. Em Capacidade máxima, insira o número máximo de fragmentos que a política do Auto Scaling do ElastiCache é solicitada a manter. Esse valor deve ser igual ou menor que 250.
13. Escolha Criar.

Registro de um destino escalável

Antes de poder usar o Auto Scaling com um cluster do ElastiCache para Valkey e Redis OSS, você deve registrar seu cluster no ajuste de escala automático do ElastiCache. Faça isso para definir a dimensão de escalabilidade e os limites a serem aplicados no cluster. O ajuste de escala automático do ElastiCache escala dinamicamente o cluster ao longo da dimensão escalável `elasticache:replication-group:NodeGroups`, que representa o número de fragmentos de cluster.

Como usar a AWS CLI

Para registrar o cluster do ElastiCache para Valkey e Redis OSS, use o comando [register-scalable-target](#) com os seguintes parâmetros:

- `--service-namespace`: defina o valor como `elasticache`
- `--resource-id`: o identificador do recurso para o cluster. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.
- `--max-capacity` : o número máximo de fragmentos a serem gerenciados pelo ajuste de escala automático do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de fragmentos em seu cluster, consulte [Capacidades mínima e máxima](#).
- `--min-capacity` : o número mínimo de fragmentos a serem gerenciados pelo ajuste de escala automático do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de fragmentos em seu cluster, consulte [Capacidades mínima e máxima](#).

Example

No exemplo a seguir, registre um cluster do ElastiCache de nome `myscalablecluster`. O registro indica que o cluster deve ser escalado dinamicamente para ter de um a dez fragmentos.

Para Linux, macOS ou Unix:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace elasticache \  
  --resource-id replication-group/myscalablecluster \  
  --scalable-dimension elasticache:replication-group:NodeGroups \  
  --min-capacity 1 \  
  --max-capacity 10 \
```

Para Windows:

```
aws application-autoscaling register-scalable-target ^  
  --service-namespace elasticache ^  
  --resource-id replication-group/myscalablecluster ^  
  --scalable-dimension elasticache:replication-group:NodeGroups ^  
  --min-capacity 1 ^  
  --max-capacity 10 ^
```

Usar a API

Para registrar o cluster do ElastiCache, use o comando [register-scalable-target](#) com os seguintes parâmetros:

- **ServiceNamespace:** defina este valor como `elasticache`.
- **ResourceID:** o identificador de recursos para o cluster do ElastiCache. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- **ScalableDimension:** defina este valor como `elasticache:replication-group:NodeGroups`.
- **MinCapacity:** o número mínimo de fragmentos a serem gerenciados pelo ajuste de escala automático do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de réplicas no cluster, consulte [Capacidades mínima e máxima](#).
- **MaxCapacity:** o número máximo de fragmentos a serem gerenciados pelo ajuste de escala automático do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de réplicas no cluster, consulte [Capacidades mínima e máxima](#).

Example

No exemplo a seguir, registre um cluster do ElastiCache chamado `myscalablecluster` na API do Application Auto Scaling. Este registro indica que o cluster deve ser escalado dinamicamente para ter de uma a 5 réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definir uma política de escalabilidade

Uma configuração de política de escalabilidade de rastreamento de destino é representada por um bloco JSON no qual as métricas e valores de destino são definidos. Você pode salvar uma configuração de política de escalabilidade como um bloco JSON em um arquivo de texto. Você pode usar esse arquivo de texto ao invocar a AWS CLI ou a API do Application Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de política, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

As seguintes opções estão disponíveis para definir uma configuração de política de escalabilidade com monitoramento do objetivo:

Tópicos

- [Uso de uma métrica predefinida](#)
- [Uso de uma métrica personalizada](#)
- [Uso de períodos de desaquecimento](#)

Uso de uma métrica predefinida

Com o uso de métricas predefinidas, você pode definir rapidamente uma política de escalabilidade com monitoramento do objetivo para um cluster do ElastiCache para Valkey e Redis OSS que funcione com o monitoramento do objetivo no ajuste de escala automático do ElastiCache.

Atualmente, o ElastiCache oferece suporte às seguintes métricas predefinidas no Auto Scaling do NodeGroup:

- `ElastiCachePrimaryEngineCPUUtilization`: o valor médio da métrica `EngineCPUUtilization` no CloudWatch em todos os nós primários no cluster.
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`: o valor médio da métrica `DatabaseMemoryUsageCountedForEvictPercentage` no CloudWatch em todos os nós primários no cluster.
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`: o valor médio da métrica `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` no CloudWatch em todos os nós primários no cluster.

Para ter mais informações sobre as métricas

`EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` e `DatabaseCapacityUsageCountedForEvictPercentage`, consulte [o uso com CloudWatch métricas](#). Para usar uma métrica predefinida em sua política de escalabilidade, crie uma configuração de rastreamento de destino para sua política de escalabilidade. Essa configuração deve incluir uma `PredefinedMetricSpecification` para a métrica predefinida e um `TargetValue` para o valor de destino dessa métrica.

Example

O exemplo a seguir descreve uma configuração de política típica para a escalabilidade com monitoramento do objetivo para um cluster do ElastiCache para Valkey e Redis OSS. Nessa configuração, a métrica predefinida `ElastiCachePrimaryEngineCPUUtilization` é usada para ajustar o cluster com base em uma utilização média da CPU de 40 por cento em todos os nós primários no cluster.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
```

```
}  
}
```

Uso de uma métrica personalizada

Com o uso de métricas personalizadas, você pode definir uma política de escalabilidade de rastreamento de destino que atenda a suas exigências personalizadas. Você pode definir uma métrica personalizada com base em qualquer métrica do ElastiCache que mude na proporção da escalabilidade. Nem todas as métricas do ElastiCache funcionam para o monitoramento do objetivo. A métrica deve ser de utilização válida e descrever o quão ocupada uma instância está. O valor da métrica deve aumentar ou diminuir na proporção do número de fragmentos no cluster. Essa aumento ou redução proporcional é necessário para usar os dados da métrica para expandir ou reduzir proporcionalmente o número de fragmentos.

Example

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, uma métrica personalizada ajusta um cluster do ElastiCache for Redis OSS com base na utilização média de uma CPU em 50 por cento em todos os fragmentos em um cluster chamado `my-db-cluster`.

```
{  
  "TargetValue": 50,  
  "CustomizedMetricSpecification":  
  {  
    "MetricName": "EngineCPUUtilization",  
    "Namespace": "AWS/ElastiCache",  
    "Dimensions": [  
      {  
        "Name": "ReplicationGroup", "Value": "my-db-cluster"  
      },  
      {  
        "Name": "Role", "Value": "PRIMARY"  
      }  
    ],  
    "Statistic": "Average",  
    "Unit": "Percent"  
  }  
}
```

Uso de períodos de desaquecimento

Você pode especificar um valor, em segundos, para `ScaleOutCooldown` a fim de adicionar um desaquecimento para expandir seu cluster. De forma similar, você pode adicionar um valor, em segundos, para `ScaleInCooldown` a fim de adicionar um período de desaquecimento para reduzir a escala na horizontal do seu cluster. Para obter mais informações, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica predefinida `ElastiCachePrimaryEngineCPUUtilization` é usada para ajustar um cluster do ElastiCache para Redis OSS com base em uma utilização média da CPU de 40 por cento em todos os nós primários do cluster. A configuração fornece um período de espera de 10 minutos para reduzir a escala horizontalmente, e um período de espera de 5 minutos para aumentar a escala horizontalmente.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "ScaleInCooldown": 600,
  "ScaleOutCooldown": 300
}
```

Desabilitar a atividade de redução

Você pode evitar que a configuração da política de escalabilidade com monitoramento de objetivo reduza a escala na horizontal de seu cluster desabilitando a atividade de redução de escala horizontal. Desabilitar a atividade de redução de escala na horizontal impede que a política de escalabilidade exclua fragmentos, enquanto permite ao mesmo tempo que a política de escalabilidade crie-os conforme necessário.

Você pode especificar um valor booleano para `DisableScaleIn` a fim de permitir ou evitar a atividade de redução de escala na horizontal no seu cluster. Para obter mais informações, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica predefinida

ElastiCachePrimaryEngineCPUUtilization ajusta um cluster do ElastiCache para Valkey e Redis OSS com base em uma utilização média da CPU de 40 por cento em todos os nós primários do cluster. A configuração desativa a atividade de redução para a política de escalabilidade.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "DisableScaleIn": true
}
```

Aplicar uma política de escalabilidade

Após registrar seu cluster com o ajuste de escala automático do ElastiCache para Valkey e Redis OSS e definir uma política de escalabilidade, aplique a política de escalabilidade ao cluster registrado. Para aplicar uma política de escalabilidade a um cluster do ElastiCache para Redis OSS, você pode usar a AWS CLI ou a API do Application Auto Scaling.

Aplicação de uma política de escalabilidade usando a AWS CLI

Para aplicar uma política de escalabilidade a seu cluster do ElastiCache para Valkey e Redis OSS, use o comando [put-scaling-policy](#) com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--policy-type`: defina este valor como `TargetTrackingScaling`.
- `--resource-id`: o identificador do recurso. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.
- `--target-tracking-scaling-policy-configuration`: a configuração da política de escalabilidade com monitoramento do objetivo a ser usada para o cluster.

No exemplo a seguir, é aplicada uma política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` a um cluster do ElastiCache para Valkey e Redis OSS chamado

`myscalablecluster` com o ajuste de escala automático do ElastiCache. Para fazer isso, use uma configuração de política salva em um arquivo chamado `config.json`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling put-scaling-policy \
  --policy-name myscalablepolicy \
  --policy-type TargetTrackingScaling \
  --resource-id replication-group/myscalablecluster \
  --service-namespace elasticache \
  --scalable-dimension elasticache:replication-group:NodeGroups \
  --target-tracking-scaling-policy-configuration file://config.json
```

Para Windows:

```
aws application-autoscaling put-scaling-policy ^
  --policy-name myscalablepolicy ^
  --policy-type TargetTrackingScaling ^
  --resource-id replication-group/myscalablecluster ^
  --service-namespace elasticache ^
  --scalable-dimension elasticache:replication-group:NodeGroups ^
  --target-tracking-scaling-policy-configuration file://config.json
```

Aplicação de uma política de escalabilidade usando a API

Para aplicar uma política de escalabilidade a seu cluster do ElastiCache para Valkey e Redis OSS, use o comando da AWS CLI [PutScalingPolicy](#) com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id`: o identificador do recurso. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.
- `--target-tracking-scaling-policy-configuration`: a configuração da política de escalabilidade com monitoramento do objetivo a ser usada para o cluster.

No exemplo a seguir, aplique a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` para um cluster do ElastiCache chamado `myscalablecluster` com a escalabilidade automática do ElastiCache. Use uma configuração de política com base na métrica predefinida `ElastiCachePrimaryEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 40.0,
    "PredefinedMetricSpecification": {
      "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    }
  }
}
```

Editar uma política de escalabilidade

Você pode editar uma política de escalabilidade usando o Console de gerenciamento da AWS, a AWS CLI ou a API do Application Auto Scaling.

Editar uma política de escalabilidade usando o Console de gerenciamento da AWS

Para editar uma política de ajuste de escala automático em um cluster do ElastiCache para Valkey e Redis OSS

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione o mecanismo apropriado.

3. Selecione o cluster ao qual você deseja adicionar uma política (escolha o botão à esquerda do nome do cluster).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Em Scaling policies (Políticas de escalabilidade), escolha o botão à esquerda da política de autoescalabilidade que você deseja alterar e selecione Modify (Modificar).
6. Faça as alterações necessárias na política.
7. Escolha Modificar.

Edição de uma política de escalabilidade usando a AWS CLI e a API

Você pode usar a AWS CLI ou a API do Application Auto Scaling para editar uma política de escalabilidade da mesma forma que você aplica uma política de escalabilidade:

- Ao usar a AWS CLI, especifique o nome da política que você deseja editar no parâmetro `--policy-name`. Especifique novos valores para os parâmetros que você deseja alterar.
- Ao usar a API do Application Auto Scaling, especifique o nome da política que você deseja editar no parâmetro `PolicyName`. Especifique novos valores para os parâmetros que você deseja alterar.

Para obter mais informações, consulte [Aplicar uma política de escalabilidade](#).

Excluir uma política de escalabilidade

Você pode excluir uma política de escalabilidade usando o Console de gerenciamento da AWS, a AWS CLI ou a API do Application Auto Scaling.

Excluir uma política de escalabilidade usando o Console de gerenciamento da AWS

Para excluir uma política do Auto Scaling de um cluster do ElastiCache para Redis OSS

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Escolha o cluster cuja política de autoescalabilidade você deseja editar (selecione o nome do cluster, e não o botão à esquerda).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).

5. Na seção Auto scaling policies (Políticas de autoescalabilidade), escolha a respectiva política e selecione Delete (Excluir).

Excluir uma política de escalabilidade usando o AWS CLI

Para excluir uma política de escalabilidade do seu cluster do ElastiCache para Valkey e Redis OSS, use o comando [delete-scaling-policy](#) AWS CLI com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id`: o identificador do recurso. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` é excluída de um cluster chamado `myscalablecluster`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling delete-scaling-policy \
  --policy-name myscalablepolicy \
  --resource-id replication-group/myscalablecluster \
  --service-namespace elasticache \
  --scalable-dimension elasticache:replication-group:NodeGroups
```

Para Windows:

```
aws application-autoscaling delete-scaling-policy ^
  --policy-name myscalablepolicy ^
  --resource-id replication-group/myscalablecluster ^
  --service-namespace elasticache ^
  --scalable-dimension elasticache:replication-group:NodeGroups
```

Exclusão de uma política de escalabilidade usando a API

Para excluir uma política de escalabilidade do seu cluster do ElastiCache para Valkey e Redis OSS, use o comando [DeleteScalingPolicy](#) AWS CLI com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id`: o identificador do recurso. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` é excluída de um cluster chamado `myscalablecluster`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups"
}
```

Use CloudFormation para políticas do Auto Scaling

Este trecho mostra como criar uma política com monitoramento do objetivo e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
```

```
MinCapacity: 1
ResourceId: !Sub replication-group/${logicalName}
ScalableDimension: 'elasticache:replication-group:NodeGroups'
ServiceNamespace: elasticache
RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization
      TargetValue: 40
```

Escalabilidade programada

A escalabilidade com base em uma programação permite que você dimensione sua aplicação em resposta a alterações de demanda. Para usar a escalabilidade programada, você deve criar ações programadas, que instruem o ElastiCache para Valkey e Redis OSS a executar ações de escalabilidade em momentos específicos. Quando você cria uma ação programada, especifica um cluster existente, quando a ação de escalabilidade deve ocorrer, a capacidade mínima e a capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

Você só pode criar uma ação programada para clusters que já existam. Você não pode criar uma ação programada ao mesmo tempo em que você cria um cluster.

Para obter mais informações sobre terminologia para criação, gerenciamento e exclusão de ações programadas, consulte [Comandos normalmente usados para criação, gerenciamento e exclusão de ações programadas](#)

Para criar em uma programação recorrente:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.

2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Escolha o cluster ao qual você deseja adicionar uma política.
4. Escolha Manage Auto Scaling policies (Gerenciar políticas do Auto Scaling) no menu suspenso Actions (Ações).
5. Escolha a guia Auto Scaling policies (Políticas do Auto Scaling).
6. Na seção Auto Scaling policies (Políticas de Auto Scaling), a caixa de diálogo Add Scaling policy (Adicionar política de escalabilidade) será exibida. Escolha Scheduled scaling (Escalabilidade programada).
7. Em Policy Name (Nome da política), insira um nome para a política.
8. Em Scalable Dimension (Dimensão escalável), escolha Shards (Fragmentos).
9. Em Target Shards (Fragmentos de destino), escolha o valor.
10. Em Recurrence (Recorrência), escolha Recurring (Recorrente).
11. Em Frequency (Frequência), escolha o respectivo valor.
12. Em Start Date (Data de início) e Start time (Horário de início), escolha o horário a partir do qual a política entrará em vigor.
13. Escolha Add policy (Adicionar política).

Para criar uma ação programada para executar uma única vez:

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Escolha o cluster ao qual você deseja adicionar uma política.
4. Escolha Manage Auto Scaling policies (Gerenciar políticas do Auto Scaling) no menu suspenso Actions (Ações).
5. Escolha a guia Auto Scaling policies (Políticas do Auto Scaling).
6. Na seção Auto Scaling policies (Políticas de Auto Scaling), a caixa de diálogo Add Scaling policy (Adicionar política de escalabilidade) será exibida. Escolha Scheduled scaling (Escalabilidade programada).
7. Em Policy Name (Nome da política), insira um nome para a política.
8. Em Scalable Dimension (Dimensão escalável), escolha Shards (Fragmentos).
9. Em Target Shards (Fragmentos de destino), escolha o valor.

10. Em Recurrence (Recorrência), escolha One Time (Uma vez).
11. Em Start Date (Data de início) e Start time (Horário de início), escolha o horário a partir do qual a política entrará em vigor.
12. Em End Date (Data de término), escolha a data até quando a política estará em vigor.
13. Escolha Add policy (Adicionar política).

Para excluir uma ação programada

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Escolha o cluster ao qual você deseja adicionar uma política.
4. Escolha Manage Auto Scaling policies (Gerenciar políticas do Auto Scaling) no menu suspenso Actions (Ações).
5. Escolha a guia Auto Scaling policies (Políticas do Auto Scaling).
6. Na seção Auto scaling policies (Políticas do Auto Scaling), escolha a política do Auto Scaling e depois escolha Delete (Excluir) na caixa de diálogo Actions (Ações).

Para gerenciar a escalabilidade programada usando o AWS CLI

Use as seguintes APIs de application-autoscaling:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Use CloudFormation para criar uma ação programada

Este trecho mostra como criar uma política com monitoramento do objetivo e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade ResourceId com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
    MinCapacity: 1
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
    ScheduledActions:
      - EndTime: '2020-12-31T12:00:00.000Z'
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
          ScheduledActionName: First
          Schedule: 'cron(0 18 * * ? *)'
```

Usar o ajuste de escala automático com réplicas

Um grupo de replicação do ElastiCache pode configurar um ou mais caches para funcionar como um único nó lógico.

O tópico seguinte fornece detalhes sobre o monitoramento do objetivo e as políticas programadas e como aplicá-las usando o Console de gerenciamento da AWS, a AWS CLI e APIs.

Políticas de escalabilidade de rastreamento de destino

Com as políticas de dimensionamento com monitoramento do objetivo, você seleciona uma métrica e define um valor pretendido. O Auto Scaling do ElastiCache para Valkey e Redis OSS cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove réplicas uniformemente ao longo de todos os fragmentos conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a um padrão de carga de flutuação, e minimiza as flutuações rápidas na capacidade da frota.

CrITÉRIOS do Auto Scaling para réplicas

Sua política do Auto Scaling define a seguinte métrica predefinida para seu cluster:

`ElastiCacheReplicaEngineCPUUtilization`: o limite de utilização de AVG EngineCPU agregado em todas as réplicas usadas pelo ElastiCache para acionar uma operação de ajuste de escala automático. Você pode definir a meta de utilização entre 35% e 70%.

Quando o serviço detectar que sua métrica `ElastiCacheReplicaEngineCPUUtilization` é igual ou maior que a configuração de destino, ele aumentará automaticamente as réplicas nos fragmentos. O ElastiCache aumenta a escala na horizontal das réplicas do seu cluster em uma contagem igual ao maior de dois números: variação percentual do objetivo e uma réplica. Para reduzir a escala horizontalmente, o ElastiCache não fará a redução da escala na horizontal automaticamente a menos que o valor geral da métrica seja inferior a 75% do seu objetivo definido.

Para um exemplo de aumento de escala na horizontal, se você tiver 5 fragmentos e 1 réplica cada:

Se o seu objetivo for violado em 30%, o ElastiCache para Valkey e Redis OSS aumentará a escala horizontalmente em 1 réplica (máximo (0,3, padrão 1)) em todos os fragmentos, o que resulta em 5 fragmentos com 2 réplicas cada.

Para um exemplo de redução de escala na horizontal, se você selecionou um valor de objetivo de 60%, o ElastiCache para Valkey e Redis OSS não reduzirá a escala na horizontal automaticamente até que a métrica seja menor ou igual a 45% (25% abaixo do objetivo de 60%).

Considerações sobre o Auto Scaling

Lembre-se das seguintes considerações:

- Uma política de escalabilidade de rastreamento de destino pressupõe que ela deve aumentar a escalabilidade quando a métrica especificada estiver acima do valor de destino. Você não pode usar uma política de escalabilidade de rastreamento de destino para expandir quando a métrica especificada estiver abaixo do valor de destino. O ElastiCache para Valkey e Redis OSS aumenta a escala na horizontal de réplicas em um máximo de (% de desvio arredondado do objetivo, padrão 1) de réplicas existentes em todos os fragmentos no cluster.
- Uma política de escalabilidade de rastreamento de destino não escala quando a métrica especificada tem dados insuficientes. Ela não aumenta a escalabilidade porque não interpreta dados insuficientes como baixa utilização.
- É possível ver lacunas entre o valor de destino e os pontos de dados de métrica reais. Isso ocorre porque o Auto Scaling do ElastiCache funciona de maneira segura por arredondamento para cima ou para baixo, quando ele determina a capacidade a ser adicionada ou removida. Isso evita que ele adicione capacidade insuficiente ou remova muita capacidade.

- Para garantir a disponibilidade da aplicação, o serviço aumenta a escala na horizontal proporcionalmente à métrica o mais rápido possível, mas reduz a escala na horizontal mais gradualmente, com redução máxima de escala de 1 réplica ao longo dos fragmentos no cluster.
- Você pode ter várias políticas de escalabilidade com monitoramento do objetivo para um cluster do ElastiCache para Valkey e Redis OSS, desde que cada uma delas use uma métrica diferente. A intenção do Auto Scaling é sempre priorizar a disponibilidade. Portanto, seu comportamento será diferente, se as políticas de monitoramento do objetivo estiverem prontas ou não para aumentar ou reduzir a escala horizontalmente. Ele vai aumentar o serviço se qualquer uma das políticas de monitoramento do objetivo estiverem prontas para aumentar, mas vai reduzir somente se todas as políticas de monitoramento do objetivo (com a parte de redução habilitada) estiverem prontas para reduzir.
- Não edite nem exclua os alarmes do CloudWatch que o ajuste de escala automático do ElastiCache gerencia para uma política de escalabilidade com monitoramento do destino. O Auto Scaling exclui os alarmes automaticamente quando você exclui a política de escalabilidade ou exclui o cluster.
- O ajuste de escala automático do ElastiCache não impede a modificação manual de réplicas nos fragmentos. Esses ajustes manuais não afetam alarmes existentes do CloudWatch que estejam conectados à política de escalabilidade, mas podem afetar métricas que possam acionar esses alarmes do CloudWatch.
- Esses alarmes do CloudWatch gerenciados pelo Auto Scaling são definidos na métrica do AVG em todos os fragmentos do cluster. Assim, ter fragmentos quentes pode resultar em qualquer cenário de:
 - escalabilidade quando não for necessário devido à carga em alguns fragmentos quentes que acionam um alarme do CloudWatch
 - não escalar quando necessário devido ao AVG agregado em todos os fragmentos que afetam o alarme não violarem.
- Os limites padrão do ElastiCache em nós por cluster ainda se aplicam. Então, ao optar pelo Auto Scaling, se você espera que os nós máximos sejam mais do que o limite padrão, solicite um aumento de limite em [Limites de serviço da AWS](#) e escolha o tipo de limite Nós por cluster por tipo de instância.
- Certifique-se de que você tenha ENIs (interfaces de rede elásticas) suficientes disponíveis em sua VPC, que são necessárias durante o aumento de escala na horizontal. Para obter mais informações, consulte [Interfaces de rede elástica](#).

- Se não houver capacidade suficiente disponível no EC2, o ajuste de escala automático do ElastiCache não aumentará a escala na horizontal até que a capacidade esteja disponível ou se você modificar manualmente o cluster para os tipos de instância que têm capacidade suficiente.
- O ajuste de escala automático do ElastiCache não oferece suporte à escalabilidade de réplicas com um cluster tendo `ReservedMemoryPercent` menos que 25%. Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

Adicionar uma política de escalabilidade

Você pode adicionar uma política de escalabilidade usando o Console de gerenciamento da AWS.

Adicionar uma política de escalabilidade usando o Console de gerenciamento da AWS

Para adicionar uma política de ajuste de escala automático ao ElastiCache para Valkey e Redis OSS

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Selecione o cluster ao qual você deseja adicionar uma política (escolha o botão à esquerda do nome do cluster).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Escolha add dynamic scaling (adicionar escalabilidade dinâmica).
6. Em Scaling Policies (Políticas de escalabilidade), escolha Add dynamic scaling (Adicionar escalabilidade dinâmica).
7. Em Policy Name (Nome da política), insira um nome para a política.
8. Em Scalable Dimension (Dimensão escalável), selecione Replicas (Réplicas) na caixa de diálogo.
9. Para o valor de destino, digite o percentual médio de utilização da CPU que você deseja manter nas réplicas do ElastiCache. Esse valor deve ser ≥ 35 e ≤ 70 . As réplicas do cluster serão adicionadas ou removidas por manter a métrica próxima ao valor especificado.
10. (Opcional) Não há suporte do console para os períodos de desaquecimento de redução ou aumento de escala na horizontal. Use a AWS CLI para modificar os valores de desaquecimento.
11. Em Minimum capacity (Capacidade mínima), insira o número mínimo de réplicas que o ajuste de escala automático do ElastiCache é solicitado a manter.

12. Em Maximum capacity (Capacidade máxima), insira o número máximo de réplicas que o ajuste de escala automático do ElastiCache é solicitado a manter. Esse valor deve ser ≥ 5 .
13. Escolha Criar.

Registro de um destino escalável

Você pode aplicar uma política de escalabilidade com base em uma métrica predefinida ou personalizada. Para fazer isso, você pode usar a AWS CLI ou a API do Application Auto Scaling. A primeira etapa é registrar seu grupo de replicação do ElastiCache para Valkey e Redis OSS no ajuste de escala automático.

Antes de poder usar o ajuste de escala automático do ElastiCache com um cluster, você deve registrar seu cluster no ajuste de escala automático do ElastiCache. Faça isso para definir a dimensão de escalabilidade e os limites a serem aplicados no cluster. O ajuste de escala automático do ElastiCache escala dinamicamente o cluster ao longo da dimensão escalável `elasticache:replication-group:Replicas`, que representa o número de réplicas por fragmento de cluster.

Usar a CLI

Para registrar o cluster do ElastiCache, use o comando [register-scalable-target](#) com os seguintes parâmetros:

- `--service-namespace`: defina este valor como `elasticache`.
- `--resource-id`: o identificador de recursos para o cluster do ElastiCache. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:Replicas`.
- `--min-capacity`: o número mínimo de fragmentos a serem gerenciados pela escalabilidade automática do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de réplicas no cluster, consulte [Capacidades mínima e máxima](#).
- `--max-capacity`: o número máximo de fragmentos a serem gerenciados pela escalabilidade automática do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de réplicas no cluster, consulte [Capacidades mínima e máxima](#).

Example

No exemplo a seguir, registre um cluster do ElastiCache de nome `myscalablecluster`. O registro indica que o cluster deve ser escalado dinamicamente para ter de uma a 5 réplicas.

Para Linux, macOS ou Unix:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace elasticache \  
  --resource-id replication-group/myscalablecluster \  
  --scalable-dimension elasticache:replication-group:Replicas \  
  --min-capacity 1 \  
  --max-capacity 5 \  

```

Para Windows:

```
aws application-autoscaling register-scalable-target ^  
  --service-namespace elasticache ^  
  --resource-id replication-group/myscalablecluster ^  
  --scalable-dimension elasticache:replication-group:Replicas ^  
  --min-capacity 1 ^  
  --max-capacity 5 ^  

```

Usar a API

Para registrar o cluster do ElastiCache, use o comando [register-scalable-target](#) com os seguintes parâmetros:

- **ServiceNamespace:** defina este valor como `elasticache`.
- **ResourceID:** o identificador de recursos para o cluster do ElastiCache. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- **ScalableDimension:** defina este valor como `elasticache:replication-group:Replicas`.
- **MinCapacity:** o número mínimo de réplicas a serem gerenciadas pela escalabilidade automática do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de réplicas no cluster, consulte [Capacidades mínima e máxima](#).
- **MaxCapacity:** o número máximo de réplicas a serem gerenciadas pela escalabilidade automática do ElastiCache. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de réplicas no cluster, consulte [Capacidades mínima e máxima](#).

Example

No exemplo a seguir, um cluster chamado `myscalablecluster` é registrado na API do Application Auto Scaling. Este registro indica que o cluster deve ser escalado dinamicamente para ter de uma a 5 réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definir uma política de escalabilidade

Uma configuração de política de escalabilidade de rastreamento de destino é representada por um bloco JSON no qual as métricas e valores de destino são definidos. Você pode salvar uma configuração de política de escalabilidade como um bloco JSON em um arquivo de texto. Você pode usar esse arquivo de texto ao invocar a AWS CLI ou a API do Application Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de política, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

As seguintes opções estão disponíveis para definir uma configuração de política de escalabilidade com monitoramento do objetivo:

Tópicos

- [Uso de uma métrica predefinida](#)
- [Editar uma política de escalabilidade](#)
- [Excluir uma política de escalabilidade](#)

- [Use CloudFormation para políticas do Auto Scaling](#)
- [Escalabilidade programada](#)

Uso de uma métrica predefinida

Uma configuração de política de escalabilidade de rastreamento de destino é representada por um bloco JSON no qual as métricas e valores de destino são definidos. Você pode salvar uma configuração de política de escalabilidade como um bloco JSON em um arquivo de texto. Você pode usar esse arquivo de texto ao invocar a AWS CLI ou a API do Application Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de política, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

As seguintes opções estão disponíveis para definir uma configuração de política de escalabilidade com monitoramento do objetivo:

Tópicos

- [Uso de uma métrica predefinida](#)
- [Uso de uma métrica personalizada](#)
- [Uso de períodos de desaquecimento](#)
- [Desabilitar a atividade de redução](#)
- [Aplicação de uma política de escalabilidade a um cluster do ElastiCache para Valkey e Redis OSS](#)

Uso de uma métrica predefinida

Com o uso de métricas predefinidas, você pode definir rapidamente uma política de escalabilidade com monitoramento do objetivo para um cluster do ElastiCache para Valkey e Redis OSS que funcione com o monitoramento do objetivo no ajuste de escala automático do ElastiCache. Atualmente, o ElastiCache dá suporte à seguinte métrica predefinida no Auto Scaling do ElastiCache:

`ElastiCacheReplicaEngineCPUUtilization`: o valor médio da métrica `EngineCPUUtilization` no CloudWatch em todas as réplicas no cluster. Você pode encontrar o valor da métrica agregada no CloudWatch em `ElastiCache ReplicationGroupId`, `Role` para `ReplicationGroupId` e réplica de função obrigatórios.

Para usar uma métrica predefinida em sua política de escalabilidade, crie uma configuração de rastreamento de destino para sua política de escalabilidade. Essa configuração deve incluir uma

`PredefinedMetricSpecification` para a métrica predefinida e um `TargetValue` para o valor de destino dessa métrica.

Uso de uma métrica personalizada

Com o uso de métricas personalizadas, você pode definir uma política de escalabilidade de rastreamento de destino que atenda a suas exigências personalizadas. Você pode definir uma métrica personalizada com base em qualquer métrica do ElastiCache para Valkey e Redis OSS que mude na proporção da escalabilidade. Nem todas as métricas do ElastiCache funcionam para o monitoramento do objetivo. A métrica deve ser de utilização válida e descrever o quão ocupada uma instância está. O valor da métrica deve aumentar ou diminuir na proporção do número de réplicas no cluster. Essa aumento ou redução proporcional é necessário para usar os dados da métrica para expandir ou reduzir proporcionalmente o número de réplicas.

Example

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, uma métrica personalizada ajusta um cluster com base em uma utilização média da CPU de 50 por cento em todas as réplicas em um cluster chamado `my-db-cluster`.

```
{
  "TargetValue": 50,
  "CustomizedMetricSpecification": {
    "MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      {
        "Name": "ReplicationGroup",
        "Value": "my-db-cluster"
      },
      {
        "Name": "Role",
        "Value": "REPLICA"
      }
    ],
    "Statistic": "Average",
    "Unit": "Percent"
  }
}
```

Uso de períodos de desaquecimento

Você pode especificar um valor, em segundos, para `ScaleOutCooldown` a fim de adicionar um desaquecimento para expandir seu cluster. De forma similar, você pode adicionar um valor, em segundos, para `ScaleInCooldown` a fim de adicionar um período de desaquecimento para reduzir a escala na horizontal do seu cluster. Para obter mais informações sobre `ScaleInCooldown` e

ScaleOutCooldown, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling. O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica predefinida `ElastiCacheReplicaEngineCPUUtilization` é usada para ajustar um cluster com base em uma utilização média da CPU de 40 por cento em todas as réplicas nesse cluster. A configuração fornece um desaquecimento de redução de 10 minutos e em um desaquecimento de expansão de 5 minutos.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
  },
  "ScaleInCooldown": 600,
  "ScaleOutCooldown": 300
}
```

Desabilitar a atividade de redução

Você pode impedir que a configuração da política de escalabilidade com monitoramento do objetivo reduza a escala horizontalmente de seu cluster do ElastiCache para Valkey e Redis OSS desabilitando a atividade de redução de escala horizontal. Desabilitar a atividade de redução de escala na horizontal impede que a política de escalabilidade exclua réplicas, enquanto permite ao mesmo tempo que a política de escalabilidade adicione-as conforme necessário.

Você pode especificar um valor booleano para `DisableScaleIn` a fim de permitir ou evitar a atividade de redução de escala na horizontal no seu cluster. Para obter mais informações sobre `DisableScaleIn`, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

Exemplo

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica predefinida `ElastiCacheReplicaEngineCPUUtilization` ajusta um cluster com base em uma utilização média da CPU de 40 por cento em todas as réplicas nesse cluster. A configuração desativa a atividade de redução para a política de escalabilidade.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
  },
  "DisableScaleIn": true
}
```

```
    },  
    "DisableScaleIn": true  
  }  
}
```

Aplicação de uma política de escalabilidade a um cluster do ElastiCache para Valkey e Redis OSS

Após registrar seu cluster com o ajuste de escala automático do ElastiCache para Valkey e Redis OSS e definir uma política de escalabilidade, aplique a política de escalabilidade ao cluster registrado. Para aplicar uma política de escalabilidade a um cluster do ElastiCache para Valkey e Redis OSS, você pode usar a AWS CLI ou a API do Application Auto Scaling.

Como usar o AWS CLI

Para aplicar uma política de escalabilidade a seu cluster do ElastiCache para Valkey e Redis OSS, use o comando [put-scaling-policy](#) com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--policy-type`: defina este valor como `TargetTrackingScaling`.
- `--resource-id`: o identificador do recurso para o cluster. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:Replicas`.
- `--target-tracking-scaling-policy-configuration`: a configuração da política de escalabilidade com monitoramento do objetivo a ser usada para o cluster.

Example

No exemplo a seguir, aplique a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` a um cluster chamado `myscalablecluster` com o ajuste de escala automático do ElastiCache. Para fazer isso, use uma configuração de política salva em um arquivo chamado `config.json`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling put-scaling-policy \  
    --policy-name myscalablepolicy \  
    --policy-type TargetTrackingScaling \  
    --resource-id replication-group/myscalablecluster \  
    --service-namespace elasticache \  
    --scalable-dimension elasticache:replication-group:Replicas \  
    --target-tracking-scaling-policy-configuration file://config.json
```



```
--policy-type TargetTrackingScaling \  
--resource-id replication-group/myscalablecluster \  
--service-namespace elasticache \  
--scalable-dimension elasticache:replication-group:Replicas \  
--target-tracking-scaling-policy-configuration file://config.json
```

```
{"TargetValue": 40.0,  
  "PredefinedMetricSpecification":  
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"  
    },  
  "DisableScaleIn": true  
}
```

Para Windows:

```
aws application-autoscaling put-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --policy-type TargetTrackingScaling ^  
  --resource-id replication-group/myscalablecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:Replicas ^  
  --target-tracking-scaling-policy-configuration file://config.json
```

Usar a API

Para aplicar a política de escalabilidade em seu cluster do ElastiCache com a API do Application Auto Scaling, use a operação [PutScalingPolicy](#) da API do Application Auto Scaling com os seguintes parâmetros:

- **PolicyName:** o nome da política de escalabilidade.
- **PolicyType:** defina este valor como `TargetTrackingScaling`.
- **ResourceID:** o identificador de recursos para o cluster. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache para Redis OSS, por exemplo `replication-group/myscalablecluster`.
- **ServiceNamespace:** defina este valor como `elasticache`.
- **ScalableDimension:** defina este valor como `elasticache:replication-group:Replicas`.
- **TargetTrackingScalingPolicyConfiguration:** a configuração da política de escalabilidade com monitoramento do objetivo a ser usada para o cluster.

Example

No exemplo a seguir, aplique a política de escalabilidade com monitoramento do objetivo chamada `scalablepolicy` a um cluster chamado `myscalablecluster` com o ajuste de escala automático do ElastiCache com Valkey ou Redis OSS. Use uma configuração de política com base na métrica predefinida `ElastiCacheReplicaEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 40.0,
    "PredefinedMetricSpecification": {
      "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    }
  }
}
```

Editar uma política de escalabilidade

Você pode editar uma política de escalabilidade usando o Console de gerenciamento da AWS, a AWS CLI ou a API do Application Auto Scaling.

Editar uma política de escalabilidade usando o Console de gerenciamento da AWS

Você só pode editar políticas com métricas do tipo Predefined (Predefinidas) usando o Console de gerenciamento da AWS

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS
3. Selecione o cluster ao qual você deseja adicionar uma política (escolha o botão à esquerda do nome do cluster).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Em Scaling policies (Políticas de escalabilidade), escolha o botão à esquerda da política de autoescalabilidade que você deseja alterar e selecione Modify (Modificar).
6. Faça as alterações necessárias na política.
7. Escolha Modificar.
8. Faça as alterações na política.
9. Escolha Modificar.

Edição de uma política de escalabilidade usando a AWS CLI ou a API do Application Auto Scaling

Você pode usar a AWS CLI ou a API do Application Auto Scaling para editar uma política de escalabilidade da mesma forma que você aplica uma política de escalabilidade:

- Ao usar a API do Application Auto Scaling, especifique o nome da política que você deseja editar no parâmetro `PolicyName`. Especifique novos valores para os parâmetros que você deseja alterar.

Para obter mais informações, consulte [Aplicação de uma política de escalabilidade a um cluster do ElastiCache para Valkey e Redis OSS](#).

Excluir uma política de escalabilidade

Você pode excluir uma política de escalabilidade usando o Console de gerenciamento da AWS, a AWS CLI ou a API do Application Auto Scaling.

Excluir uma política de escalabilidade usando o Console de gerenciamento da AWS

Você só pode editar políticas com métricas do tipo Predefined (Predefinidas) usando o Console de gerenciamento da AWS

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.

2. No painel de navegação, escolha Valkey ou Redis OSS
3. Escolha o cluster cuja política do Auto Scaling você deseja excluir.
4. Escolha a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Na seção Auto scaling policies (Políticas de autoescalabilidade), escolha a respectiva política e selecione Delete (Excluir).

Exclusão de uma política de escalabilidade usando a AWS CLI ou a API do Application Auto Scaling

Você pode usar a AWS CLI ou a API do Application Auto Scaling para excluir uma política de escalabilidade de um cluster do ElastiCache.

CLI

Para excluir uma política de escalabilidade do seu cluster do ElastiCache para Valkey e Redis OSS, use o comando [delete-scaling-policy](#) com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id`: o identificador do recurso para o cluster. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:Replicas`.

Example

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` é excluída de um cluster do ElastiCache chamado `myscalablecluster`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling delete-scaling-policy \  
  --policy-name myscalablepolicy \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:Replicas \  
  --region us-east-1
```

Para Windows:

```
aws application-autoscaling delete-scaling-policy ^
  --policy-name myscalablepolicy ^
  --resource-id replication-group/myscalablecluster ^
  --service-namespace elasticache ^
  --scalable-dimension elasticache:replication-group:Replicas ^
```

API

Para excluir uma política de escalabilidade do seu cluster do ElastiCache para Valkey e Redis OSS, use a operação da API do Application Auto Scaling [DeleteScalingPolicy](#) com os seguintes parâmetros:

- **PolicyName:** o nome da política de escalabilidade.
- **ResourceID:** o identificador de recursos para o cluster. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster, por exemplo `replication-group/myscalablecluster`.
- **ServiceNamespace:** defina este valor como `elasticache`.
- **ScalableDimension:** defina este valor como `elasticache:replication-group:Replicas`.

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` é excluída de um cluster chamado `myscalablecluster` com a API do Application Auto Scaling.

```
POST / HTTP/1.1
>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas"
```

```
}
```

Use CloudFormation para políticas do Auto Scaling

Este trecho mostra como criar uma ação programada e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
    ScalableDimension: 'elasticache:replication-group:Replicas'
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization
      TargetValue: 40
```

Escalabilidade programada

A escalabilidade com base em uma programação permite que você dimensione sua aplicação em resposta a alterações de demanda. Para usar a escalabilidade programada, você deve criar ações programadas, que instruem o ElastiCache para Valkey e Redis OSS a executar ações de escalabilidade em momentos específicos. Ao criar uma ação programada, você especifica um cluster existente do ElastiCache, quando a ação de escalabilidade deve ocorrer, a capacidade mínima e a

capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

Você só pode criar uma ação programada para clusters do ElastiCache que já existam. Você não pode criar uma ação programada ao mesmo tempo em que você cria um cluster.

Para obter mais informações sobre terminologia para criação, gerenciamento e exclusão de ações programadas, consulte [Comandos normalmente usados para criação, gerenciamento e exclusão de ações programadas](#)

Para criar uma ação programada para executar uma única vez:

Semelhante à dimensão do fragmento. Consulte [Escalabilidade programada](#).

Para excluir uma ação programada

Semelhante à dimensão do fragmento. Consulte [Escalabilidade programada](#).

Para gerenciar a escalabilidade programada usando o AWS CLI

Use as seguintes APIs de application-autoscaling:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Use CloudFormation para criar políticas do Auto Scaling

Este trecho mostra como criar uma ação programada e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
```

```
MinCapacity: 0
ResourceId: !Sub replication-group/${logicalName}
ScalableDimension: 'elasticache:replication-group:Replicas'
ServiceNamespace: elasticache
RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
ScheduledActions:
  - EndTime: '2020-12-31T12:00:00.000Z'
    ScalableTargetAction:
      MaxCapacity: '5'
      MinCapacity: '2'
      ScheduledActionName: First
      Schedule: 'cron(0 18 * * ? *)'
```

Modificar o modo de cluster

O Valkey e o Redis OSS são bancos de dados em memória distribuídos que oferecem suporte à fragmentação e à replicação. Os clusters do ElastiCache Valkey e Redis OSS são a implementação distribuída que permite que os dados sejam particionados em vários nós. Um cluster do ElastiCache para Redis OSS tem dois modos de operação: o modo de cluster ativado (CME) e o modo de cluster desativado (CMD). No CME, um mecanismo Valkey e Redis OSS funciona como um banco de dados distribuído com vários fragmentos e nós, enquanto no CMD, o Valkey e o Redis OSS funcionam como um único nó.

Antes de migrar do CMD para o CME, as seguintes condições devem ser atendidas:

Important

A configuração do modo de cluster só pode ser alterada do modo de cluster desativado para o modo cluster ativado. Não é possível reverter essa configuração.

- O cluster só pode ter chaves no banco de dados 0.
- As aplicações devem usar um cliente Valkey ou Redis OSS que seja capaz de usar o protocolo Cluster e um endpoint de configuração.
- O failover automático deve ser ativado no cluster com no mínimo uma réplica.
- A versão mínima do mecanismo necessária para a migração é o Valkey 7.2 e posteriores ou o Redis OSS 7.0 e posteriores.

Para migrar do CMD para o CME, a configuração do modo de cluster deve ser alterada do modo de cluster desativado para o modo cluster ativado. Esse é um procedimento de duas etapas que garante a disponibilidade do cluster durante o processo de migração.

 Note

Você precisa fornecer um grupo de parâmetros com configuração habilitada para cluster, ou seja, o parâmetro habilitado para cluster deve estar definido como `yes`. Se você estiver usando um grupo de parâmetros padrão, o ElastiCache para Redis OSS escolherá automaticamente o grupo de parâmetros padrão correspondente com uma configuração habilitada para cluster. O valor do parâmetro habilitado para cluster é definido como `no` para um cluster CMD. Conforme o cluster passa para o modo compatível, o valor do parâmetro habilitado para cluster é atualizado para `yes` como parte da ação de modificação.

Para obter mais informações, consulte [Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache](#).

1. Preparar-se: crie um cluster CME de teste e certifique-se de que sua pilha esteja pronta para trabalhar com ele. O ElastiCache para Redis OSS não tem como verificar sua prontidão. Para obter mais informações, consulte [Criação de um cluster do Valkey ou Redis OSS](#).
2. Modificar a configuração existente do cluster CMD para compatível com o modo cluster: nesse modo, haverá um único fragmento implantado e o ElastiCache para Redis OSS funcionará como um único nó, mas também como um único cluster de fragmento. O modo compatível significa que o aplicativo cliente pode usar qualquer um dos protocolos para se comunicar com o cluster. Nesse modo, as aplicações devem ser reconfigurados para começar a usar o protocolo de cluster do Valkey ou do Redis OSS e o endpoint de configuração. Para alterar o modo cluster do Valkey ou do Redis OSS para o modo cluster compatível, siga as etapas abaixo:

 Note

No modo compatível, outras operações de modificação, como ajuste de escala e versionamento do mecanismo, não são permitidas para o cluster. Além disso, os parâmetros (exceto `cacheParameterGroupName`) não podem ser modificados ao definir o parâmetro do modo de cluster na solicitação [ModifyReplicationGroup](#).

- a. Usando o Console de gerenciamento da AWS, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Compatível
- b. Usando a API, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `ClusterMode` para `compatible`.
- c. Usando a AWS CLI, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `cluster-mode` para `compatible`.

Depois de alterar o modo cluster do Valkey ou Redis OSS para o modo cluster compatível, a API [DescribeReplicationGroups](#) retornará o endpoint de configuração do cluster ElastiCache para Redis OSS. O endpoint de configuração do cluster é um endpoint único que pode ser usado pelos aplicativos para se conectar ao cluster. Para obter mais informações, consulte [Encontrar endpoints de conexão no ElastiCache](#).

3. Modificar a configuração do cluster para o modo cluster ativado: depois que o modo de cluster for definido como compatível com o modo de cluster, a segunda etapa é modificar a configuração do cluster para o modo cluster ativado. Nesse modo, um único fragmento está em execução, e os clientes agora podem ajustar a escala dos seus clusters ou modificar outras configurações de cluster.

Para alterar o modo de cluster para ativado, siga estas etapas:

Antes de começar, certifique-se de que seus clientes Valkey ou Redis OSS tenham migrado para o protocolo de cluster e que o endpoint de configuração do cluster não esteja em uso.

- a. Usando o Console de gerenciamento da AWS, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Habilitado.
- b. Usando a API, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `ClusterMode` para `enabled`.
- c. Usando a AWS CLI, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `cluster-mode` para `enabled`.

Depois de alterar o modo cluster para habilitado, os endpoints serão configurados de acordo com a especificação do cluster do Valkey ou Redis OSS. A API [DescribeReplicationGroups](#) retornará o parâmetro do modo de cluster como `enabled` e os endpoints do cluster que agora estão disponíveis para serem usados pelos aplicativos para se conectar ao cluster.

Observe que os endpoints do cluster mudarão quando o modo de cluster for alterado para ativado. Certifique-se de atualizar seus aplicativos com os novos endpoints.

Você também pode optar por voltar ao modo de cluster desativado (CMD) usando o modo de cluster compatível e preservar as configurações originais.

Modificar a configuração do cluster para o modo de cluster desativado usando o modo de cluster compatível

1. Usando o Console de gerenciamento da AWS, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Desabilitado.
2. Usando a API, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `ClusterMode` para `disabled`.
3. Usando a AWS CLI, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `cluster-mode` para `disabled`.

Depois de alterar o modo de cluster para desativado, a API [DescribeReplicationGroups](#) retornará o parâmetro do modo de cluster como `disabled`.

Replicação entre AWS regiões usando datastores globais

Note

No momento, o Global Datastore só está disponível para clusters baseados em nós.

Ao usar o recurso Global Datastore, você pode trabalhar com a replicação de clusters Valkey ou Redis OSS totalmente gerenciada, rápida, confiável e segura em todas as regiões. AWS Usando esse recurso, você pode criar clusters de réplicas de leitura entre regiões para permitir leituras de baixa latência e recuperação de desastres em todas as regiões. AWS

Nas seções a seguir, você pode encontrar uma descrição de como trabalhar com datastores globais.

Tópicos

- [Visão geral do](#)
- [Pré-requisitos e limitações](#)

- [Uso de datastores globais \(console\)](#)
- [Uso de datastores globais \(CLI\)](#)

Visão geral do

Cada datastore global é uma coleção de um ou mais clusters que são replicados entre si.

Um datastore global consiste no seguinte:

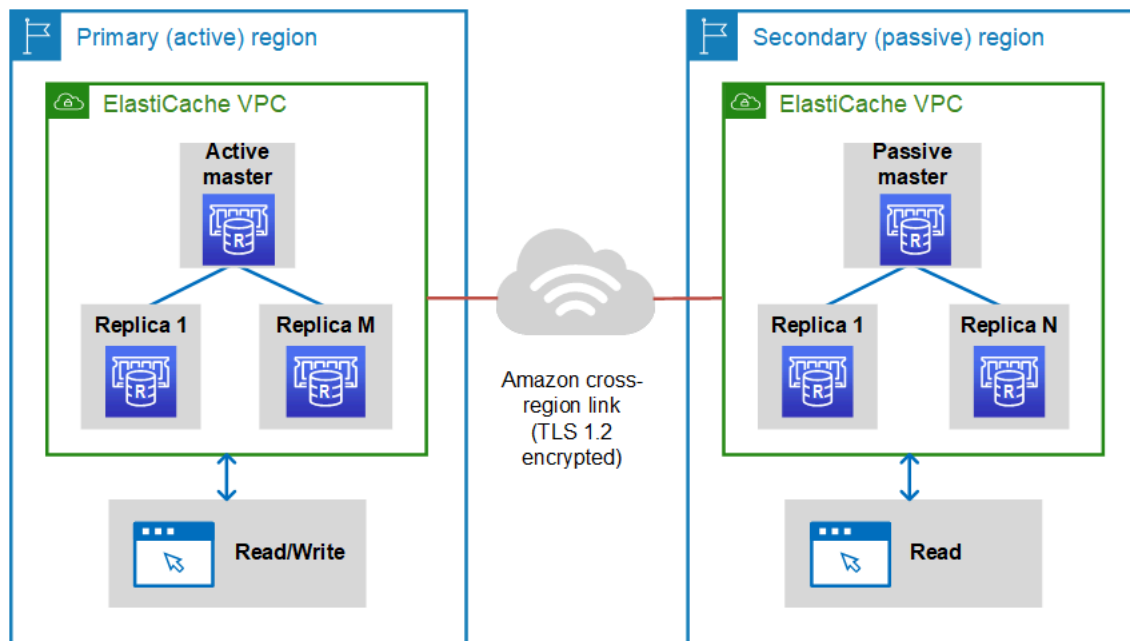
- Cluster primário (ativo): um cluster primário aceita gravações que são replicadas para todos os clusters dentro do datastore global. Um cluster primário também aceita solicitações de leitura.
- Cluster secundário (passivo): um cluster secundário só aceita solicitações de leitura e replica as atualizações de dados de um cluster primário. Um cluster secundário precisa estar em uma AWS região diferente do cluster primário.

Quando você cria um armazenamento de dados global ElastiCache para Valkey ou Redis OSS, ele replica automaticamente seus dados do cluster primário para o cluster secundário. Você escolhe a AWS região em que os dados do Valkey ou do Redis OSS devem ser replicados e, em seguida, cria um cluster secundário nessa região. AWS ElastiCache em seguida, configura e gerencia a replicação automática e assíncrona de dados entre os dois clusters.

O uso de um datastore global para Valkey ou Redis OSS oferece as seguintes vantagens:

- Desempenho geolocal — ao configurar clusters de réplicas remotas em AWS regiões adicionais e sincronizar seus dados entre elas, você pode reduzir a latência do acesso aos dados nessa região. AWS Um armazenamento de dados global pode ajudar a aumentar a capacidade de resposta do seu aplicativo fornecendo leituras geolocais de baixa latência em todas as regiões. AWS
- Recuperação de desastres se o cluster primário em um datastore global sofrer degradação, você poderá promover um cluster secundário como seu novo cluster primário. Você pode fazer isso conectando-se a qualquer AWS região que contenha um cluster secundário.

O diagrama a seguir mostra como os datastores globais podem funcionar.



Pré-requisitos e limitações

Ao começar a usar datastores globais, lembre-se do seguinte:

- Os datastores globais são compatíveis com as seguintes regiões da AWS:
 - África: Cidade do Cabo
 - Ásia-Pacífico: Hong Kong, Hyderabad, Jacarta, Malásia, Melbourne, Mumbai, Osaka, Seul, Cingapura, Sydney, Tailândia e Tóquio
 - Canadá: Central do Canadá e Oeste do Canadá (Calgary)
 - China: Pequim e Ningxia
 - Europa: Frankfurt, Londres, Irlanda, Milão, Paris, Espanha, Estocolmo e Zurique
 - AWS GovCloud: Oeste dos EUA e Leste dos EUA
 - Israel: Tel Aviv
 - Oriente Médio: Bahrein e Emirados Árabes Unidos
 - Leste dos EUA (Norte da Virgínia e Ohio) e Oeste dos EUA (Norte da Califórnia e Oregon)
 - América do Sul: México (Centro) e São Paulo
- Todos os clusters (primário e secundário) datastore global devem ter o mesmo número de nós primários, tipo de nó, versão do mecanismo e número de fragmentos (caso o modo cluster esteja habilitado). Cada cluster no datastore global pode ter um número diferente de réplicas de leitura para acomodar o tráfego de leitura local para esse cluster.

A replicação deve ser habilitada se você planejar usar um cluster de nó único existente.

- Os datastores globais são compatíveis com instâncias de tamanho grande ou superior.
- É possível configurar a replicação de um cluster primário de uma região da AWS para um cluster secundário em até duas outras regiões da AWS.

 Note

A exceção a isso são regiões China (Pequim) e China (Ningxia), onde a replicação só pode ocorrer entre as duas regiões.

- É possível trabalhar com datastores globais somente em clusters de VPC. Para obter mais informações, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#). Os datastores globais não são compatíveis ao usar o EC2-Classik. Para obter mais informações, consulte [EC2-Classik](#) no Guia do usuário do Amazon EC2.

 Note

No momento, você não pode usar datastores globais no [Uso de zonas locais com o ElastiCache](#).

- O ElastiCache não oferece suporte ao failover automático de uma região da AWS para outra. Quando necessário, você pode promover um cluster secundário manualmente. Para obter um exemplo, consulte [Promoção do cluster secundário para primário](#).
- Para a inicialização com os dados existentes, use um cluster existente como primário para criar um datastore global. Não oferecemos suporte à adição de um cluster existente como secundário. O processo de adição do cluster como secundário limpa os dados, o que pode resultar em perda de dados.
- As atualizações de parâmetros são aplicadas a todos os clusters quando você modifica um grupo de parâmetros local de um cluster pertencente a um datastore global.
- Você pode dimensionar clusters regionais verticalmente (dimensionando para cima e para baixo) e horizontalmente (reduzindo e ampliando). Você pode dimensionar os clusters modificando o datastore global. Todos os clusters regionais no datastore global são depois dimensionados sem interrupção. Para obter mais informações, consulte [Dimensionamento ElastiCache](#).
- Os datastores globais oferecem suporte à [criptografia em repouso](#), à [criptografia em trânsito](#) e ao [AUTH](#).

- Os datastores globais não oferecem suporte ao Protocolo de Internet versão 6 (IPv6).
- Suporte a chaves de AWS KMS de datastores globais. Para obter mais informações, consulte [Conceitos do serviço de gerenciamento de chaves da AWS](#) no Guia do desenvolvedor do AWS Key Management Service.

Note

Os datastores globais oferecem suporte ao [sistema de mensagens pub/sub](#) com as seguintes estipulações:

- No modo de cluster desabilitado, pub/sub é totalmente compatível. Os eventos publicados no cluster primário da região primária da AWS são propagados para regiões secundárias da AWS.
- No modo de cluster habilitado, o seguinte se aplica:
 - Para eventos publicados que não estejam em um espaço de chaves, apenas os assinantes na mesma região da AWS recebem os eventos.
 - Em eventos de espaço de chaves publicados, os assinantes em todas as regiões da AWS recebem esses eventos.

Uso de datastores globais (console)

Para criar um datastore global usando o console, siga este processo de duas etapas:

1. Crie um cluster primário, usando um cluster existente ou criando um cluster. O mecanismo deve ser o Valkey 7.2 ou posterior ou o Redis OSS 5.0.6 ou posterior.
2. Adicione até dois clusters secundários em diferentes regiões da AWS, novamente usando o Valkey 7.2 ou posterior ou o Redis OSS 5.0.6 ou posterior.

Os procedimentos a seguir orientam você a criar um datastore global do Valkey ou Redis OSS e executar outras operações usando o console do ElastiCache.

Tópicos

- [Criar um datastore global usando um cluster existente](#)
- [Criação de um datastore global usando um novo cluster primário](#)
- [Visualização de detalhes do datastore global](#)

- [Adição de uma região a um datastore global](#)
- [Modificação de um datastore global](#)
- [Promoção do cluster secundário para primário](#)
- [Remoção de uma região de um datastore global](#)
- [Exclusão de um datastore global](#)

Criar um datastore global usando um cluster existente

Neste cenário, você usa um cluster existente para servir como o primário do novo datastore global. Em seguida, você cria um cluster secundário somente leitura em uma região da AWS separada. Esse cluster secundário recebe atualizações automáticas e assíncronas do cluster primário.

Important

O cluster existente deve usar um mecanismo Valkey 7.2 ou posterior ou o Redis OSS 5.0.6 ou posterior.

Como criar um datastore global usando um cluster existente

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais e, em seguida, escolha Criar datastore global.
3. Na página Configurações do cluster primário, faça o seguinte:
 - No campo Informações do datastore global, insira um nome para o novo datastore global.
 - (Opcional) Insira um valor para Description (Descrição).
4. Em Cluster regional, selecione Usar cluster regional existente.
5. Em Cluster existente, selecione o cluster existente que você deseja usar.
6. Mantenha as seguintes opções como estão. Elas são pré-preenchidas para corresponder à configuração do cluster primário, você não pode alterá-las.
 - Versão do mecanismo
 - Tipo de nó

- Grupo de parâmetros

 Note

O ElastiCache gera automaticamente um grupo de parâmetros com os valores do grupo de parâmetros fornecido e aplica o novo grupo de parâmetros ao cluster. Use esse novo grupo de parâmetros para modificar parâmetros em um datastore global. Cada grupo de parâmetros gerado automaticamente está associado a um, e apenas a um, cluster e, portanto, apenas a um datastore global.

- Número de fragmentos
- Encryption at rest (Criptografia em repouso): permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

 Note

Se deseja fornecer uma chave de criptografia diferente, selecione Chave do AWS KMS gerenciada pelo cliente e escolha a chave. Para mais informações, consulte [Uso de chaves do AWS KMS gerenciadas pelo cliente](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [Criptografia em trânsito](#). Para o Valkey 7.2 e posteriores, e Redis OSS versão 6.0 e posteriores, ao habilitar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Controle de acesso:
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições.
 - User Group Access Control List (Lista de controle de acesso do grupo de usuários): escolha um grupo de usuários com um conjunto definido de usuários e permissões nas operações disponíveis. Para obter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
 - Usuário padrão AUTH: um mecanismo de autenticação para um servidor Valkey ou Redis OSS. Para obter mais informações, consulte [AUTH](#).
7. (Opcional) Conforme necessário, atualize as configurações restantes do cluster secundário. Elas são pré-preenchidas com os mesmos valores das do cluster primário, mas você pode atualizá-las para atender aos requisitos específicos para esse cluster.
- Porta

- Número de réplicas
 - Grupo de sub-rede
 - Zona(s) de disponibilidade preferencial(is)
 - Grupos de segurança
 - Gerenciada pelo cliente (chave do KMS da AWS)
 - Token AUTH
 - Habilitar backups automáticos
 - Backup retention period (Período de retenção de backup)
 - Janela de backup
 - Maintenance window
 - Tópico para notificação do SNS
8. Escolha Criar. Fazer isso define o status do datastore global como Creating (Criando). As transições de status para Modifying (Modificando) depois que o cluster primário é associado ao datastore global e o cluster secundário está no status Associating (Associando).

Depois que o cluster primário e os clusters secundários estão associados ao datastore global, o status muda para Available (Disponível). Neste ponto, você tem um cluster primário, que aceita leituras e gravações, e clusters secundários, que aceitam leituras replicadas do cluster primário.

A página é atualizada para indicar se um cluster faz parte de um datastore global, incluindo:

- Global Datastore (Datastore global): o nome datastore global ao qual o cluster pertence.
- Global Datastore Role (Função do datastore global): a função do cluster, primário ou secundário.

Você pode adicionar até um cluster secundário a uma região da AWS diferente. Para obter mais informações, consulte [Adição de uma região a um datastore global](#).

Criação de um datastore global usando um novo cluster primário

Se você optar por criar um datastore global com um novo cluster, use o procedimento a seguir.

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais e, em seguida, escolha Criar datastore global.

3. Em Primary cluster settings (Configurações do cluster primário), faça o seguinte:
 - a. Em Cluster mode (Modo cluster), escolha Enabled (Habilitado) ou Disabled (Desabilitado).
 - b. Em Global Datastore info (Informações do datastore global), insira um valor para Name (Nome). O ElastiCache usa o sufixo para gerar um nome exclusivo para o datastore global. Você pode pesquisar o datastore global usando o sufixo especificado aqui.
 - c. (Opcional) Insira um valor para Global Datastore Description (Descrição do datastore global).
4. Em Regional cluster (Cluster regional):
 - a. Em Region (Região), escolha uma Região da AWS disponível.
 - b. Selecione Create new regional cluster (Criar cluster regional) ou Use existing regional cluster (Usar cluster regional existente)
 - c. Se escolher Create new regional cluster (Criar cluster regional), em Cluster info (Informações sobre o cluster), insira um nome e uma descrição opcional do cluster.
 - d. Em Location (Localização), recomendamos que você aceite as configurações padrão de Multi-AZ e Auto-failover (Failover automático).
5. Em Cluster settings (Configurações do cluster)
 - a. Em Engine version (Versão do mecanismo), escolha uma versão disponível, que é 5.0.6 ou posterior.
 - b. Em Port (Porta), use a porta padrão, 6379. Se você tiver um motivo para usar uma porta diferente, insira o número da porta.
 - c. Em Grupo de parâmetro (Parameter group), escolha um grupo de parâmetro ou crie um novo. Os grupo de parâmetros controlam os parâmetros de tempo de execução do seu cluster. Para obter mais informações sobre grupo de parâmetros, consulte [Parâmetros do Valkey e do Redis OSS](#) e [Criação de um grupo de parâmetros do ElastiCache](#).

 Note

Quando você seleciona um grupo de parâmetros para definir os valores de configuração do mecanismo, esse grupo de parâmetros é aplicado a todos os clusters no datastore global. Na página Parameter Groups (Grupos de parâmetros), o atributo Global indica se um grupo de parâmetros faz parte de um datastore global.

- d. Para Node type (Tipo de nó), escolha a seta para baixo (▼)).

Na caixa de diálogo Change node type (Alterar tipo de nó), escolha um valor para a Instance family (Família de instâncias) para o tipo de nó desejado. Depois disso, escolha o tipo de nó que deseja usar para este cluster e escolha Save (Salvar).

Para obter mais informações, consulte [Escolha do tamanho do nó](#).

Se você escolher um tipo de nó r6gd, a classificação de dados em níveis será ativada automaticamente. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

- e. Se você estiver criando um cluster do Valkey ou Redis OSS (modo cluster desabilitado):

Em Number of replicas (Número de réplicas), escolha o número de réplicas desejado para esse cluster.

- f. Se você estiver criando um cluster do Valkey ou Redis OSS (modo cluster habilitado):

- i. Em Número de fragmentos, escolha o número de fragmentos (partições/grupos de nós) desejado para este cluster do Valkey ou Redis OSS (modo cluster habilitado).

Para algumas versões do Valkey ou Redis OSS (modo cluster habilitado), é possível alterar dinamicamente o número de fragmentos no cluster:

- Redis OSS 3.2.10 e posteriores: se o cluster estiver executando o Redis OSS 3.2.10 ou versões posteriores, será possível alterar dinamicamente o número de fragmentos no cluster. Para obter mais informações, consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).
- Outras versões do Redis OSS: se o cluster estiver executando uma versão do Redis OSS anterior à versão 3.2.10, há outra abordagem. Para alterar o número de fragmentos no cluster nesse caso, crie um novo cluster com o novo número de fragmentos. Para obter mais informações, consulte [Restauração de um backup para um novo cache](#).

- ii. Em Réplicas por fragmento, escolha o número de nós de réplica de leitura desejados em cada fragmento.

As restrições a seguir existem para o Valkey ou Redis OSS (modo cluster habilitado).

- Se você tiver o Multi-AZ habilitado, verifique se tem pelo menos uma réplica por fragmento.
 - O número de réplicas é o mesmo para cada fragmento ao criar o cluster usando o console.
 - O número de réplicas de leitura por fragmento é fixo e não pode ser alterado. Se você achar que precisa de mais ou menos réplicas por fragmento (API/CLI: grupo de nós), deverá criar um novo cluster com o novo número de réplicas. Para obter mais informações, consulte [Tutorial: propagação de um novo cluster baseado em nós com um backup criado externamente](#).
6. Em Subnet group settings (Configurações de grupo de sub-redes), escolha a sub-rede que você deseja aplicar a esse cluster. O ElastiCache fornece um grupo de sub-redes IPv4 padrão ou é possível optar por criar um novo. Para o IPv6, é necessário criar um grupo de sub-rede com um bloco CIDR do IPv6. Se você escolher dual stack (pilha dupla), deverá selecionar um tipo de IP de descoberta: IPv6 ou IPv4.

Para obter mais informações, consulte [Criar uma sub-rede na sua VPC](#).

7. Em Availability zone placements (Posicionamentos de zona de disponibilidade), você tem duas opções:
- No preference (Sem preferência): o ElastiCache escolhe a zona de disponibilidade.
 - Specify availability zones (Especificar zonas de disponibilidade): você especifica a zona de disponibilidade para cada cluster.

Se optar por especificar as zonas de disponibilidade, para cada cluster em cada fragmento, escolha a zona de disponibilidade na lista.

Para obter mais informações, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#).

Slots and keyspaces: Custom distribution

Availability zone(s): Specify availability zones

	Slots/Keyspaces	Primary	Replica 1
NodeGroup 1	0-1234	us-east-1a	us-east-1a
NodeGroup 2		us-east-1b	us-east-1a
NodeGroup 3		us-east-1a	us-east-1a

Especificar espaços de chaves e zonas de disponibilidade

8. Escolha Next (Próximo)

9. Em Configurações avançadas do Valkey e do Redis OSS

- Para Security (Segurança):
 - i. Para criptografar seus dados, você tem as seguintes opções:
 - Criptografia em repouso: permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

Note

Você tem a opção de fornecer uma chave de criptografia diferente, escolhendo Customer Managed KMS key (Chave do AWS KMS gerenciada pelo cliente) e escolhendo a chave. Para obter mais informações, consulte [Usar chaves gerenciadas pelo cliente do AWS KMS](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [criptografia em trânsito](#). Para o Valkey 7.2 e posteriores e Redis OSS 6.0 e posteriores, se você habilitar a Criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Controle de acesso:

- No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.
- User Group Access Control List (Lista de controle de acesso de grupo de usuários): selecione um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para obter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
- Usuário padrão AUTH: um mecanismo de autenticação para um servidor Valkey ou Redis OSS. Para obter mais informações, consulte [AUTH](#).
- AUTH: um mecanismo de autenticação para um servidor Valkey ou Redis OSS. Para obter mais informações, consulte [AUTH](#).

 Note

Para versões do Redis OSS 3.2.6 e posteriores, excluindo a versão 3.2.10, o AUTH é a única opção.

- ii. Em Grupos de segurança, escolha os grupos de segurança desejados para esse cluster. Um grupo de segurança atua como um firewall para controlar o acesso à rede ao cluster. É possível usar o grupo de segurança padrão para sua VPC ou criar um novo.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança para sua VPC](#) no Guia do usuário da Amazon VPC.

10. Para backups automáticos agendados regularmente, selecione Enable automatic backups (Habilitar backups automáticos) e insira o número de dias que deseja que cada backup automático seja mantido antes de ser excluído automaticamente. Se não quiser backups automáticos agendados regularmente, desmarque a caixa de seleção Enable automatic backups (Habilitar backups automáticos). Em ambos os casos, você sempre tem a opção de criar backups manuais.

Para obter mais informações sobre backup e restauração, consulte [Snapshots e restauração](#).

11. (Opcional) Especifique uma janela de manutenção. A maintenance window (janela de manutenção) é o tempo, geralmente uma hora de duração, a cada semana quando o ElastiCache programa a manutenção do sistema para seu cluster. É possível permitir que o ElastiCache escolha o dia e a hora da sua janela de manutenção (No preference (Sem preferência)) ou é possível escolher o dia, a hora e a duração por conta própria (Specify

maintenance window (Especificar janela de manutenção)). Se você escolher Especificar janela de manutenção, nas listas, escolha Dia de início, Hora de início e Duração (em horas) para sua janela de manutenção. Todos os horários são em UCT.

Para obter mais informações, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

12. (Opcional) Para Logs:

- Em Log format (Formato do log), escolha Text (Texto) ou JSON.
- Em Destination Type (Tipo de destino), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.
- Em Destino do log, escolha Criar novo e insira o nome do grupo de logs do CloudWatch Logs ou o nome do stream do Firehose, ou escolha Selecionar existente e escolha o nome do grupo de logs do CloudWatch Logs ou o nome do stream do Firehose,

13. Em tags, para ajudar você a gerenciar seus clusters e outros recursos do ElastiCache, é possível atribuir seus próprios metadados a cada recurso na forma de tags. Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).

14. Revise todas as suas entradas e opções e faça as correções necessárias. Quando estiver pronto, selecione Avançar.

15. Depois de configurar o cluster nas etapas anteriores, agora você vai configurar os detalhes do cluster secundário.

16. Em Regional cluster (Cluster regional), escolha a Região da AWS onde o cluster está localizado.

17. Em Cluster info (Informações sobre o cluster), insira um nome e uma descrição opcional do cluster.

18. As opções seguintes são pré-preenchidas para corresponder à configuração do cluster primário e não podem ser alteradas:

- Local
- Versão do mecanismo
- Tipo de instância
- Tipo de nó
- Número de fragmentos
- Grupo de parâmetros

 Note

O ElastiCache gera automaticamente um grupo de parâmetros com os valores do grupo de parâmetros fornecido e aplica o novo grupo de parâmetros ao cluster. Use esse novo grupo de parâmetros para modificar parâmetros em um datastore global. Cada grupo de parâmetros gerado automaticamente está associado a um, e apenas a um, cluster e, portanto, apenas a um datastore global.

- Encryption at rest (Criptografia em repouso): permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

 Note

Se deseja fornecer uma chave de criptografia diferente, selecione Chave do AWS KMS gerenciada pelo cliente e escolha a chave. Para mais informações, consulte [Uso de chaves do AWS KMS gerenciadas pelo cliente](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [Criptografia em trânsito](#). Para o Valkey 7.2 e posteriores e Redis OSS 6.4 e posteriores, se você habilitar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Controle de acesso:
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.
 - User Group Access Control List (Lista de controle de acesso de grupo de usuários): escolha um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para obter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
 - Usuário padrão AUTH: um mecanismo de autenticação para um servidor Valkey ou Redis OSS. Para obter mais informações, consulte [AUTH](#).

 Note

Para versões do Redis OSS entre 4.0.2, quando houve suporte para a criptografia em trânsito pela primeira vez, e 6.0.4, o AUTH é a única opção.

As configurações de cluster secundário restantes são pré-preenchidas com os mesmos valores que o cluster primário, mas o seguinte pode ser atualizado para atender aos requisitos específicos para esse cluster:

- Porta
- Número de réplicas
- Grupo de sub-rede
- Zona(s) de disponibilidade preferencial(is)
- Grupos de segurança
- Gerenciada pelo cliente (chave do KMS da AWS)
- Token AUTH
- Habilitar backups automáticos
- Backup retention period (Período de retenção de backup)
- Janela de backup
- Maintenance window
- Tópico para notificação do SNS

19. Escolha Create. Isso define o status do datastore global como Creating (Criando). Depois que o cluster primário e os clusters secundários estão associados ao datastore global, o status muda para Available (Disponível). Você tem um cluster primário que aceita leituras e gravações e um cluster secundário que aceita leituras replicadas do cluster primário.

A página também é atualizada para indicar se um cluster faz parte de um datastore global, incluindo o seguinte:

- Global Datastore (Datastore global): o nome datastore global ao qual o cluster pertence.
- Global Datastore Role (Função do datastore global): a função do cluster, primário ou secundário.

Você pode adicionar até um cluster secundário a uma região da AWS diferente. Para obter mais informações, consulte [Adição de uma região a um datastore global](#).

Visualização de detalhes do datastore global

Você pode visualizar os detalhes dos datastores globais existentes e também modificá-los na página [Datastores globais](#).

Como visualizar os detalhes do datastore global

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais e escolha um datastore global disponível.

Depois, você pode examinar as seguintes propriedades do datastore global:

- Global Datastore Name: o nome do datastore global
- Descrição: uma descrição do datastore global
- Status: as opções incluem:
 - Criando
 - Modificar
 - Available (Disponível)
 - Excluindo
 - Somente primário - esse status indica que o datastore global contém apenas um cluster primário. Todos os clusters secundários são excluídos ou não são criados com êxito.
- Modo de cluster: habilitado ou desabilitado
- Versão do mecanismo: a versão do mecanismo Valkey ou Redis OSS que executa o datastore global
- Tipo de nó de instância: o tipo de nó usado para o datastore global
- Criptografia em repouso: habilitada ou desabilitada
- Criptografia em trânsito: Habilitada ou desabilitada
- AUTH: habilitada ou desabilitada

Você pode fazer as seguintes alterações no datastore global:

- [Adição de uma região a um datastore global](#)
- [Remoção de uma região de um datastore global](#)

- [Promoção do cluster secundário para primário](#)
- [Modificação de um datastore global](#)

A página Global Datastore (Datastore global) também lista os clusters individuais que compõem o datastore global e as seguintes propriedades de cada um:

- Region (Região): a região da AWS em que o cluster está armazenado
- Role (Função) - primária ou secundária
- Cluster name (Nome do cluster) - o nome do cluster
- Status - as opções incluem:
 - Associating (Associando) - o cluster está em processo de ser associado ao datastore global
 - Associated (Associado) - o cluster está associado ao datastore global
 - Disassociating (Desassociando) - o processo de remoção de um cluster secundário do datastore global que usa o nome do datastore global. Depois disso, o cluster secundário não recebe mais atualizações do cluster primário, mas permanece como um cluster independente nessa região da AWS.
 - Disassociated (Desassociado): o cluster secundário foi removido do datastore global e agora é um cluster independente em sua região da AWS.
- Global Datastore Replica lag (Atraso da réplica do datastore global): mostra um valor por região secundária da AWS no datastore global. Esse é o atraso entre o nó primário da região secundária e o nó primário da região primária. Para o Valkey ou Redis OSS habilitado para modo cluster, o atraso indica o atraso máximo, em segundos, entre os fragmentos.

Adição de uma região a um datastore global

Você pode adicionar até uma região adicional da AWS a um datastore global existente. Nesse cenário, você está criando um cluster somente leitura em uma região separada da AWS que recebe atualizações automáticas e assíncronas do cluster primário.

Para adicionar uma região da AWS a um datastore global

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais e selecione um datastore global existente.

3. Escolha Adicionar cluster regional e escolha a região da AWS em que o cluster secundário deve residir.
4. Em Informações do cluster, insira um valor para Nome e, opcionalmente, para Descrição do cluster.
5. Mantenha as seguintes opções como estão. Elas são pré-preenchidas para corresponder à configuração do cluster primário, e você não pode alterá-las.

- Versão do mecanismo
- Tipo de instância
- Tipo de nó
- Número de fragmentos
- Grupo de parâmetros

 Note

O ElastiCache gera automaticamente um grupo de parâmetros com os valores do grupo de parâmetros fornecido e aplica o novo grupo de parâmetros ao cluster. Use esse novo grupo de parâmetros para modificar parâmetros em um datastore global. Cada grupo de parâmetros gerado automaticamente está associado a um, e apenas a um, cluster e, portanto, apenas a um datastore global.

- Criptografia em repouso

 Note

Se deseja fornecer uma chave de criptografia diferente, selecione Chave do AWS KMS gerenciada pelo cliente e escolha a chave.

- Criptografia em trânsito
 - AUTH
6. (Opcional) Atualize as configurações restantes do cluster secundário. Elas são pré-preenchidas com os mesmos valores das do cluster primário, mas é possível atualizá-las para atender aos requisitos específicos para esse cluster:
 - Porta
 - Número de réplicas

- Grupo de sub-rede
- Zona(s) de disponibilidade preferencial(is)
- Grupos de segurança
- Chave do KMS da AWS gerenciada pelo cliente
- Token AUTH
- Habilitar backups automáticos
- Backup retention period (Período de retenção de backup)
- Janela de backup
- Maintenance window
- Tópico para notificação do SNS

7. Escolha Add (Adicionar).

Modificação de um datastore global

É possível modificar propriedades de clusters regionais. Somente uma operação de modificação pode estar em andamento em um datastore global, com exceção da promoção de um cluster secundário para primário. Para obter mais informações, consulte [Promoção do cluster secundário para primário](#).

Como modificar um datastore global

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais e, em Nome do datastore global, escolha um datastore global.
3. Escolha Modify (Modificar) e escolha entre as seguintes opções:
 - Modify description (Modificar descrição): atualiza a descrição do datastore global
 - Modificar versão do mecanismo: somente o Valkey 7.2 e posteriores ou Redis OSS 5.0.6 e posteriores estão disponíveis.
 - Modify node type (Modificar tipo de nó): escale clusters regionais tanto verticalmente (aumentando e reduzindo a escala vertical) quanto horizontalmente (aumentando e reduzindo a escala horizontal). As opções incluem as famílias de nós R5 e M5. Para obter mais informações sobre tipos de nós, consulte [Tipos de nó compatíveis](#).

- **Modify Automatic Failover (Modificar failover automático):** ativa ou desativa o failover automático. Quando você habilita o failover e os nós primários em clusters regionais são encerrados inesperadamente, o ElastiCache faz failover para uma das réplicas regionais. Para obter mais informações, consulte [Failover automático](#).

Para clusters do Valkey ou Redis OSS com modo cluster habilitado:

- **Add shards (Adicionar fragmentos):** insira o número de fragmentos a serem adicionados e, opcionalmente, especifique uma ou mais zonas de disponibilidade.
- **Delete shards (Excluir fragmentos):** escolha os fragmentos a serem excluídos em cada região da AWS.
- **Rebalance shards (Rebalancear fragmentos):** reequilibre a distribuição do slot para garantir uma distribuição uniforme entre fragmentos existentes no cluster.

Para modificar os parâmetros de um datastore global, modifique o grupo de parâmetros de qualquer cluster membro para o datastore global. O ElastiCache aplica essa alteração a todos os clusters dentro desse datastore global automaticamente. Para modificar o grupo de parâmetros desse cluster, use o console do Valkey ou do Redis OSS ou a operação da API [ModifyCacheCluster](#). Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#). Quando você modifica o grupo de parâmetros de qualquer cluster contido em um datastore global, ele é aplicado a todos os clusters dentro desse datastore global.

Para redefinir um grupo de parâmetros inteiro ou parâmetros específicos, use a operação de API [ResetCacheParameterGroup](#).

Promoção do cluster secundário para primário

Se o cluster primário ou a região da AWS se tornar indisponível ou estiver enfrentando problemas de performance, você poderá promover um cluster secundário para primário. A promoção é permitida a qualquer momento, mesmo que outras modificações estejam em andamento. Você também pode emitir várias promoções em paralelo e o datastore global será resolvido para um primário eventualmente. Se você promover vários clusters secundários simultaneamente, o ElastiCache não garantirá qual deles será resolvido como primário.

Como promover um cluster secundário para primário

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.

2. No painel de navegação, escolha Datastores globais.
3. Selecione o nome do datastore global para visualizar os detalhes
4. Escolha o cluster Secondary (Secundário)
5. Escolha Promote to primary (Promover para principal).

Depois, você será solicitado a confirmar sua decisão com o seguinte aviso: Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the *secondary* cluster to primary?

The current primary cluster in *primary region* will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.

6. Escolha Confirm (Confirmar) se quiser continuar a promoção ou, caso contrário, escolha Cancel (Cancelar).

Se você optar por confirmar, o datastore global será movido para um estado Modifying (Modificando) e ficará indisponível até que a promoção seja concluída.

Remoção de uma região de um datastore global

Você pode remover uma região da AWS de um datastore global usando o procedimento a seguir.

Como remover uma região da AWS de um datastore global

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais.
3. Escolha um datastore global.
4. Escolha a Region (Região) que deseja remover.
5. Escolha Remove region (Remover região).

Note

Essa opção só está disponível para clusters secundários.

Depois, você será solicitado a confirmar sua decisão com o seguinte aviso: Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore?

6. Escolha Confirm (Confirmar) se quiser continuar a promoção ou, caso contrário, escolha Cancel (Cancelar).

Se você escolher confirmar, a região da AWS será removida, e o cluster secundário não receberá mais atualizações de replicação.

Exclusão de um datastore global

Para excluir um datastore global, primeiro remova todos os clusters secundários. Para obter mais informações, consulte [Remoção de uma região de um datastore global](#). Fazer isso deixa o datastore global no status primary-only (somente primário).

Como excluir um datastore global

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Datastores globais.
3. Em Global Datastore Name (Nome do datastore global) escolha o datastore global desejado e escolha Delete (Excluir).

Depois, você será solicitado a confirmar sua decisão com o seguinte aviso: Are you sure you want to delete this Global Datastore?

4. Escolha Excluir.

O datastore global faz a transição para o status Deleting (Excluindo).

Uso de datastores globais (CLI)

Você pode usar a AWS Command Line Interface (AWS CLI) para controlar vários serviços da AWS via linha de comando e automatizá-los usando scripts. Você pode usar a CLI da AWS para operações ad hoc (únicas).

Download e configuração da AWS CLI

A AWS CLI é executada no Windows, no macOS ou no Linux. Use o procedimento a seguir para fazer download e configurá-la.

Como fazer download, instalar e configurar a CLI

1. Baixe a CLI da AWS na página [Interface de linha de comando da AWS](#).
2. Siga as instruções para instalação da CLI da AWS e configurar a CLI da AWS no Guia do usuário da AWS Command Line Interface.

Uso da CLI da AWS com datastores globais

Use as seguintes operações da CLI para trabalhar com datastores globais:

- [create-global-replication-group](#)

```
aws elasticache create-global-replication-group \
  --global-replication-group-id-suffix my global datastore \
  --primary-replication-group-id sample-repl-group \
  --global-replication-group-description an optional description of the global
datastore
```

O Amazon ElastiCache aplica automaticamente um prefixo ao ID do datastore global quando ele é criado. Cada região da AWS tem seu próprio prefixo. Por exemplo, um ID de datastore global criado na região Oeste dos EUA (Norte da Califórnia) começa com "virxk" junto com o nome do sufixo fornecido. O sufixo, combinado com o prefixo gerado automaticamente, garante a exclusividade do nome do datastore global em várias regiões.

A tabela a seguir lista cada região da AWS e seu prefixo de ID de datastore global.

Nome da região/região	Prefixo
Região Leste dos EUA (Ohio)	fpkhr
us-east-2	
Região Leste dos EUA (Norte da Virgínia)	ldgnf

Nome da região/região	Prefixo
us-east-1	
Região Oeste dos EUA (Norte da Califórnia)	virxk
us-west-1	
Região Oeste dos EUA (Oregon)	sgau1
us-west-2	
Região Canadá (Central)	bxodz
ca-central-1	
Região Ásia-Pacífico (Mumbai)	erpgt
ap-south-1	
Região Ásia-Pacífico (Tóquio)	quwsw
ap-northeast-1	
Região Ásia-Pacífico (Seul)	1fqnh
ap-northeast-2	
Região Ásia-Pacífico (Osaka)	n1apn
ap-northeast-3	
Região Ásia-Pacífico (Singapura)	v1qxn
ap-southeast-1	
Região Ásia-Pacífico (Sydney)	vbgxd
ap-southeast-2	

Nome da região/região	Prefixo
Região Europa (Frankfurt) eu-central-1	iudkw
Região Europa (Irlanda) eu-west-1	gxeiz
Região Europa (Londres) eu-west-2	okuqm
Região Europa (Paris) eu-west-3	fgjhi
Região América do Sul (São Paulo) sa-east-1	juxlw
Região da China (Pequim) cn-north-1	emvgo
Região da China (Ningxia) cn-northwest-1	ckbem
Região Ásia-Pacífico (Hong Kong) ap-east-1	knjmp
AWS GovCloud (Oeste dos EUA) us-gov-west-1	sgwui

- [create-replication-group](#): use esta operação para criar clusters secundários para um datastore global fornecendo o nome do datastore global ao parâmetro `--global-replication-group-id`.

```
aws elasticache create-replication-group \  
  --replication-group-id secondary replication group name \  
  --replication-group-description "Replication group description" \  
  --global-replication-group-id global datastore name
```

Ao chamar esta operação e passar um valor `--global-replication-group-id`, o ElastiCache irá inferir os valores do grupo de replicação principal do grupo de replicação global para os seguintes parâmetros. Não passe valores para estes parâmetros:

"PrimaryClusterId",
"AutomaticFailoverEnabled",
"NumNodeGroups",
"CacheParameterGroupName",
"CacheNodeType",
"Engine",
"EngineVersion",
"CacheSecurityGroupNames",
"EnableTransitEncryption",
"AtRestEncryptionEnabled",
"SnapshotArns",
"SnapshotName"

- [describe-global-replication-groups](#)

```
aws elasticache describe-global-replication-groups \  
  --global-replication-group-id my global datastore \  
  --show-member-info an optional parameter that returns a list of the primary and  
  secondary clusters that make up the global datastore
```

- [modify-global-replication-group](#)

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --automatic-failover-enabled \  
  --cache-node-type node type \  
  --cache-parameter-group-name parameter group name \  
  --engine-version engine version \  
  --apply-immediately \  
  --global-replication-group-description description
```

Atualização entre mecanismos do Redis OSS para Valkey para ElastiCache GlobalDataStore

Você pode atualizar um grupo de replicação global do Redis OSS existente para o Valkey usando o Console, a API ou a CLI.

Se você tiver um grupo de replicação global do Redis OSS existente, poderá fazer a atualização para o Valkey especificando o novo mecanismo e a versão do mecanismo com a API `modify-global-replication-group`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id myGlobalReplGroup \  
  --engine valkey \  
  --apply-immediately \  
  --engine-version 8.0
```

Para Windows:

```
aws elasticache modify-global-replication-group ^  
  --global-replication-group-id myGlobalReplGroup ^  
  --engine valkey ^  
  --apply-immediately ^  
  --engine-version 8.0
```

Se você tiver um grupo de parâmetros de cache personalizado aplicado ao grupo de replicação global do Redis OSS existente que deseja atualizar, também será necessário passar um grupo de parâmetros de cache personalizado do Valkey na solicitação. O grupo de parâmetros

personalizados Valkey de entrada deve ter os mesmos valores de parâmetros estáticos do Redis OSS que o grupo de parâmetros personalizados do Redis OSS existente.

Para Linux, macOS ou Unix:

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id myGlobalReplGroup \  
  --engine valkey \  
  --engine-version 8.0 \  
  --apply-immediately \  
  --cache-parameter-group-name myParamGroup
```

Para Windows:

```
aws elasticache modify-global-replication-group ^  
  --global-replication-group-id myGlobalReplGroup ^  
  --engine valkey ^  
  --engine-version 8.0 ^  
  --apply-immediately ^  
  --cache-parameter-group-name myParamGroup
```

- [delete-global-replication-group](#)

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --retain-primary-replication-group defaults to true
```

- [disassociate-global-replication-group](#)

```
aws elasticache disassociate-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --replication-group-id my secondary cluster \  
  --replication-group-region the AWS Region in which the secondary cluster resides
```

- [failover-global-replication-group](#)

```
aws elasticache failover-replication-group \  
  --global-replication-group-id my global datastore \  
  --primary-region The AWS Region of the primary cluster \  
  --primary-replication-group-id The name of the global datastore, including the suffix.
```

- [increase-node-groups-in-global-replication-group](#)

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [decrease-node-groups-in-global-replication-group](#)

```
aws elasticache decrease-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [rebalance-shards-in-global-replication-group](#)

```
aws elasticache rebalance-shards-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name
```

Use a ajuda para listar todos os comandos disponíveis para o ElastiCache para Valkey ou Redis OSS.

```
aws elasticache help
```

Você também pode usar a ajuda para descrever um comando específico e saber mais sobre seu uso:

```
aws elasticache create-global-replication-group help
```

Alta disponibilidade com o uso de grupos de replicação

Os clusters do Valkey e Redis OSS do Amazon ElastiCache de nó único são entidades na memória com serviços de proteção de dados limitados (AOF). Se o seu cluster falhar por qualquer motivo, você perderá todos os dados do cluster. No entanto, se você estiver executando o mecanismo Valkey ou Redis OSS, poderá agrupar de 2 a 6 nós em um cluster com réplicas em que 1 a 5 nós somente leitura contêm dados replicados do único nó primário de leitura/gravação do grupo. Nesse cenário, se um nó falhar por qualquer motivo, você não perderá todos os seus dados, pois eles serão

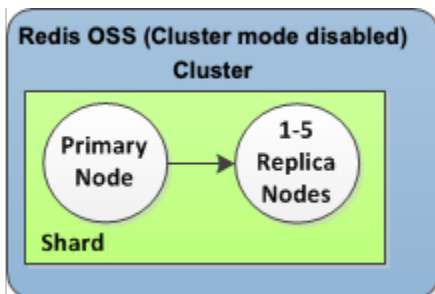
replicados em um ou mais outros nós. Devido à latência de replicação, alguns dados podem ser perdidos se a falha ocorrer no nó primário de leitura/gravação.

Conforme visto no gráfico a seguir, a estrutura de replicação está contida dentro de um fragmento (chamado de grupo de nós na API/CLI) que está contido dentro de um cluster do Valkey ou Redis OSS. Clusters do Valkey ou Redis OSS (modo cluster desabilitado) sempre têm um fragmento. Clusters do Valkey ou Redis OSS (modo cluster habilitado) podem ter até 500 fragmentos, com os dados do cluster particionados nos fragmentos. É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster com o ElastiCache para Valkey e com o ElastiCache versão 5.0.6 ou superior para Redis OSS. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).



Cluster do Valkey ou Redis OSS (modo cluster desabilitado) tem um fragmento e de 0 a 5 nós de réplicas

Se o cluster com réplicas tiver o Multi-AZ habilitado e o nó primário falhar, esse nó primário executará failover em uma réplica de leitura. Como os dados são atualizados nos nós de réplica de forma assíncrona, pode haver alguma perda de dados devido à latência na atualização dos nós de réplica. Para obter mais informações, consulte [Mitigar falhas ao executar o Valkey ou Redis OSS](#).

Tópicos

- [Entendendo a replicação do Valkey e do Redis OSS](#)
- [Replicação: Modo de cluster Valkey e Redis OSS desativado versus ativado](#)
- [Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#)
- [Como a sincronização e o backup são implementados](#)
- [Criação de um grupo de replicação do Valkey ou Redis OSS](#)
- [Visualização dos detalhes de um grupo de replicação](#)
- [Localização de endpoints de grupo de replicação](#)
- [Modificação de um grupo de replicação](#)
- [Exclusão de um grupo de replicação](#)
- [Alteração do número de réplicas](#)
- [Promoção de uma réplica de leitura a primário, para grupos de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\)](#)

Entendendo a replicação do Valkey e do Redis OSS

O Redis OSS implementa a replicação de duas formas:

- Com um único fragmento que contém todos os dados do cluster em cada nó: Valkey ou Redis OSS (modo cluster desabilitado)
- Com dados particionados em até 500 fragmentos: Valkey ou Redis OSS (modo cluster habilitado)

Cada fragmento em um grupo de replicação tem um único nó primário de leitura/gravação e até 5 nós de réplica somente de leitura. É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo Redis OSS for 5.0.6 ou superior. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

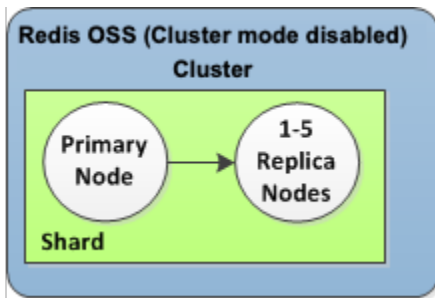
Tópicos

- [Valkey ou Redis OSS \(modo cluster desabilitado\)](#)
- [Valkey ou Redis OSS \(modo cluster habilitado\)](#)

Valkey ou Redis OSS (modo cluster desabilitado)

Um cluster do Valkey ou Redis OSS (modo cluster desabilitado) tem um único fragmento, dentro do qual há uma coleção de nós; um nó de leitura/gravação primário e até cinco nós de réplica secundários somente de leitura. Cada réplica de leitura mantém uma cópia dos dados do nó primário do cluster. Mecanismos de replicação assíncronos são usados para manter as réplicas de leitura sincronizadas com a primária. Os aplicativos podem ler a partir de qualquer nó no cluster. Os

aplicativos podem apenas gravar no nó primário. Réplicas de leitura melhoram o throughput de leitura e protegem contra a perda de dados nos casos de falha de um nó.



Cluster do Valkey ou Redis OSS (modo cluster desabilitado) com um único fragmento e nós de réplicas

Você pode usar clusters do Valkey ou Redis OSS (modo cluster desabilitado) com réplicas de nós para escalar sua solução para o ElastiCache de modo a lidar com aplicações de leitura intensiva ou para oferecer suporte a um grande número de clientes que leiam simultaneamente do mesmo cluster.

Todos os nós em um cluster do Valkey ou Redis OSS (modo cluster desabilitado) devem residir na mesma região.

Quando você adiciona uma réplica de leitura a um cluster, todos os dados do primário são copiados para o novo nó. A partir desse momento, sempre que os dados são gravados no primário, as alterações são propagadas de forma assíncrona em todas as réplicas de leitura.

Para melhorar a tolerância a falhas e reduzir o tempo de gravação, habilitar o recurso multi-AZ com failover automático para o seu cluster do Valkey ou Redis OSS (modo cluster desabilitado) com réplicas. Para obter mais informações, consulte [Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#).

Você pode alterar as funções dos nós dentro do cluster do Valkey ou Redis OSS (modo cluster desabilitado) com o primário e uma das réplicas trocando funções. Você pode decidir fazer isso por motivos de ajuste de desempenho. Por exemplo, com um aplicativo Web que possui uma intensa atividade de gravação, você pode escolher o nó que tenha a menor latência da rede. Para obter mais informações, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\)](#).

Valkey ou Redis OSS (modo cluster habilitado)

Um cluster do Valkey ou Redis OSS (modo cluster habilitado) é composto de 1 a 500 fragmentos (API/CLI: grupos de nós). Cada fragmento tem um nó primário e até cinco nós de réplica de leitura.

A configuração pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

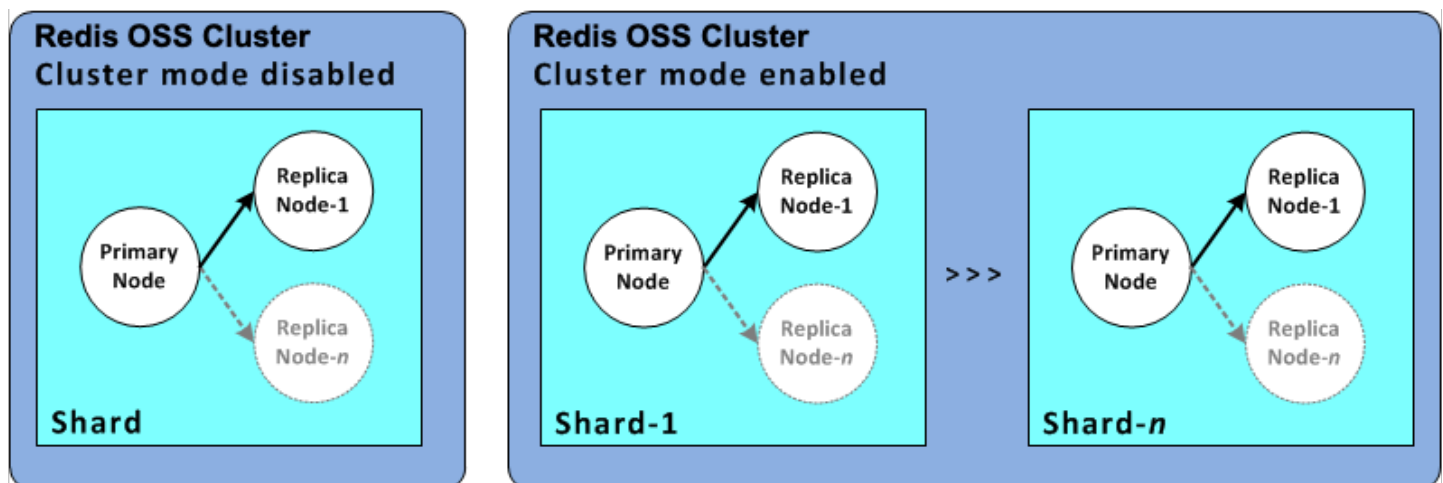
O limite de nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo for Valkey 7.2 e posteriores ou Redis OSS 5.0.6 e posteriores. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Cada réplica de leitura em um fragmento mantém uma cópia dos dados do primário do fragmento. Mecanismos de replicação assíncronos são usados para manter as réplicas de leitura sincronizadas com a primária. Os aplicativos podem ler a partir de qualquer nó no cluster. Os aplicativos podem apenas gravar nos nós primários. Réplicas de leitura melhoram a escalabilidade de leitura e protegem contra a perda de dados. Os dados são particionados em todos os fragmentos em um cluster do Valkey ou Redis OSS (modo cluster habilitado).

As aplicações usam o endpoint de configuração do cluster do Valkey ou Redis OSS (modo cluster habilitado) para se conectar aos nós no cluster. Para obter mais informações, consulte [Encontrar endpoints de conexão no ElastiCache](#).



Cluster do Valkey ou Redis OSS (modo cluster habilitado) com vários fragmentos e nós de réplica

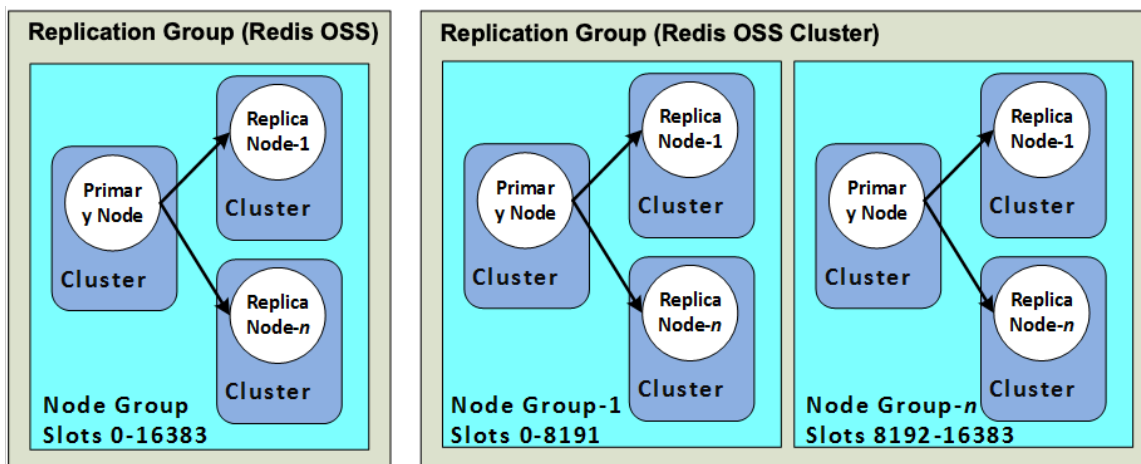
Todos os nós em um cluster do Valkey ou Redis OSS (modo cluster habilitado) devem residir na mesma região. Para melhorar a tolerância a falhas, você pode provisionar primários e réplicas de leitura em várias zonas de disponibilidade dentro dessa região.

Atualmente, os recursos Valkey ou Redis OSS (modo cluster habilitado) têm algumas limitações.

- Você não pode promover manualmente nenhum nó de réplica para primário.

Replicação: Modo de cluster Valkey e Redis OSS desativado versus ativado

Começando com o Valkey 7.2 e o Redis OSS versão 3.2, você pode criar um dos dois tipos distintos de clusters (API/CLI: grupos de replicação). Um cluster do Valkey ou Redis OSS (modo cluster desabilitado) sempre possui um único fragmento (API/CLI: grupo de nós) com até 5 nós de réplicas de leitura. Um cluster do Valkey ou Redis OSS (modo cluster habilitado) tem até 500 fragmentos de 1 a 5 nós de réplicas de leitura em cada.



Clusters do Valkey ou Redis OSS (modo cluster desabilitado) e do Redis OSS (modo cluster habilitado)

A tabela a seguir resume diferenças importantes entre os clusters do Valkey ou Redis OSS (modo cluster desabilitado) e do Valkey ou Redis OSS (modo cluster habilitado).

Comparação de clusters do Valkey ou Redis OSS (modo cluster desabilitado) e do Valkey ou Redis OSS (modo cluster habilitado)

Recurso	Valkey ou Redis OSS (modo cluster desabilitado)	Valkey ou Redis OSS (modo cluster habilitado)
Modificável	Sim. Oferece suporte para adicionar e excluir nós de réplica e aumentar a escala verticalmente do tipo de nó.	Limitado. Para obter mais informações, consulte Gerenciamento de versões para ElastiCache e Escalabilidade de clusters no Valkey ou Redis OSS (modo cluster habilitado) .
Particionamento de dados	Não	Sim
Estilhaços	1	1 a 500
Réplicas de leitura	0 – 5  Important Se você não tiver réplicas e o nó falhar, ocorrerá uma perda total de dados.	0 a 5 por fragmento.  Important Se você não tiver réplicas e um nó falhar, ocorrerá uma perda de todos os dados nesse estilhaço.
Multi-AZ	Sim, com pelo menos 1 réplica. Opcional. Ativado por padrão.	Sim Opcional. Ativado por padrão.
Snapshots (Backups)	Sim, criando um único arquivo .rdb.	Sim, criando um arquivo .rdb exclusivo para cada estilhaço.
Restaurar	Sim, usando um único arquivo .rdb de um cluster do	Sim, usando arquivos .rdb de um cluster do Valkey ou Redis OSS (modo cluster desabilitado).

Recurso	Valkey ou Redis OSS (modo cluster desabilitado)	Valkey ou Redis OSS (modo cluster habilitado)
	Valkey ou Redis OSS (modo cluster desabilitado).	ado) ou do Valkey ou Redis OSS (modo cluster habilitado).
Com suporte por	Todas as versões do Valkey e Redis OSS	Todas as versões do Valkey e Redis OSS 3.2 e seguintes
Mecanismo atualizável	Sim, com alguns limites. Para obter mais informações, consulte Gerenciamento de versões para ElastiCache .	Sim, com alguns limites. Para obter mais informações, consulte Gerenciamento de versões para ElastiCache .
Criptografia	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e 4.0.10 e posteriores.	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e 4.0.10 e posteriores.
Elegível para HIPAA	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e 4.0.10 e posteriores.	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e 4.0.10 e posteriores.
Compatível com PCI DSS	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e 4.0.10 e posteriores.	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e 4.0.10 e posteriores.
Reestilhecimento online	N/D	Versão 3.2.10 (programa da para EOL, consulte Cronograma de fim de vida útil das versões do Redis OSS) e posterior.

Qual devo escolher?

Ao escolher entre Valkey ou Redis OSS (modo cluster desabilitado) ou Valkey ou Redis OSS (modo cluster habilitado), considere os seguintes fatores:

- Escalabilidade versus particionamento: os negócios precisam mudar. Você precisa fornecer provisão para demanda máxima ou escalar conforme a demanda sofre alterações. O Valkey ou Redis OSS (modo cluster desabilitado) oferece suporte a escalabilidade. Você pode dimensionar a capacidade de leitura adicionando ou excluindo nós de réplica ou pode dimensionar a capacidade aumentando para um tipo de nó maior. Ambas as operações exigem tempo. Para obter mais informações, consulte [Escalar nós de réplica para Valkey ou Redis OSS \(modo cluster desativado\)](#).

O Valkey ou Redis OSS (modo cluster habilitado) oferece suporte para o particionamento dos seus dados em até 500 grupos de nós. Você pode alterar dinamicamente o número de estilhaços à medida que suas necessidades de negócios mudam. Uma vantagem do particionamento é que você espalha sua carga em um número maior de endpoints que reduz o afunilamento de acesso durante a demanda de pico. Além disso, você pode acomodar um conjunto de dados maior, pois os dados podem ser distribuídos em vários servidores. Para obter informações sobre como escalar suas partições, consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).

- Tamanho do nó versus número de nós: como um cluster do Valkey ou Redis OSS (modo cluster desabilitado) tem apenas um fragmento, o tipo de nó deve ser grande o suficiente para acomodar todos os dados do cluster mais a sobrecarga necessária. Por outro lado, como você pode particionar seus dados em vários fragmentos ao usar um cluster do Valkey ou Redis OSS (modo cluster habilitado), os tipos de nós podem ser menores, embora você precise de mais deles. Para obter mais informações, consulte [Escolha do tamanho do nó](#).
- Leituras versus gravações: se a carga principal no seu cluster for de aplicações que leiam dados, você poderá escalar um cluster do Valkey ou Redis OSS (modo cluster habilitado) adicionando e excluindo réplicas de leitura. No entanto, há um máximo de 5 réplicas de leitura. Se a carga no seu cluster for de escrita pesada, você poderá se beneficiar dos endpoints de gravação adicionais de um cluster do Valkey ou Redis OSS (modo cluster habilitado) com vários fragmentos.

Seja qual for o tipo de cluster que você optar por implementar, certifique-se de escolher um tipo de nó que seja adequado para suas necessidades atuais e futuras.

Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS

Há várias instâncias em que o ElastiCache para Valkey e Redis OSS pode precisar substituir um nó primário; elas incluem certos tipos de manutenção planejada e o evento improvável de falha em um nó primário ou zona de disponibilidade.

Essa substituição resulta em algum tempo de inatividade do cluster, mas se o multi-AZ estiver habilitado, o tempo de inatividade será minimizado. A função do nó primário fará failover automaticamente para uma das réplicas de leitura. Não há necessidade de criar e provisionar um novo nó primário, porque o ElastiCache lidará com isso de forma transparente. O failover e a promoção de réplica garantem que você possa continuar a gravar no novo primário assim que a promoção estiver concluída.

O ElastiCache também propaga o nome do DNS (Serviço de Nomes de Domínio) da réplica promovida. Isso ocorre porque, em seguida, se o seu aplicativo estiver gravando no endpoint primário, nenhuma alteração nesse endpoint será necessária no seu aplicativo. Se estiver lendo de endpoints individuais, altere o endpoint de leitura da réplica promovida a primária para o endpoint da nova réplica.

No caso de substituições de nó planejadas, iniciadas devido a atualizações de manutenção ou atualizações de autoatendimento, esteja ciente do seguinte:

- Para clusters do Valkey e Redis OSS, as substituições de nó planejadas são concluídas enquanto o cluster atende às solicitações de gravação recebidas.
- Para clusters desabilitados do modo de cluster do Valkey e Redis OSS com o Multi-AZ habilitado que são executados no mecanismo 5.0.6 ou posterior, as substituições de nó planejadas são concluídas enquanto o cluster atende às solicitações de gravação recebidas.
- Para clusters desabilitados do modo de cluster do Valkey e Redis OSS com o Multi-AZ habilitado que são executados no mecanismo 4.0.10 ou anterior, é possível notar uma breve interrupção de gravação associada às atualizações de DNS. Essa interrupção pode levar até alguns segundos. Esse processo é muito mais rápido do que recriar e provisionar um novo primário, que será o caso se você não habilitar o multi-AZ.

É possível habilitar o multi-AZ usando o console de gerenciamento do ElastiCache, a AWS CLI ou a API do ElastiCache.

Habilitar o Multi-AZ do ElastiCache no cluster do Valkey ou Redis OSS (na API e CLI ou no grupo de replicação) melhora a tolerância a falhas. Isso é verdade, especialmente nos casos em que o cluster primário de leitura/gravação do seu cluster se torna inacessível ou falha por qualquer motivo. O Multi-AZ só é aceito em clusters do Valkey e Redis OSS com mais de um nó em cada fragmento.

Tópicos

- [Habilitar Multi-AZ](#)
- [Cenários de falha com respostas do multi-AZ](#)
- [Teste do failover automático](#)
- [Limitações no Multi-AZ](#)

Habilitar Multi-AZ

É possível habilitar o multi-AZ ao criar ou modificar um cluster (API, CLI ou grupo de replicação) usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

É possível habilitar o Multi-AZ somente em clusters do Valkey ou Redis OSS (modo cluster desabilitado) que têm pelo menos uma réplica de leitura disponível. Clusters sem réplicas de leitura não fornece alta disponibilidade ou tolerância a falhas. Para obter informações sobre como criar um cluster com replicação, consulte [Criação de um grupo de replicação do Valkey ou Redis OSS](#). Para obter informações sobre como adicionar uma réplica de leitura a um cluster com replicação, consulte [Adição de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#).

Tópicos

- [Habilitação do multi-AZ \(console\)](#)
- [Habilitar o recurso multi-AZ \(AWS CLI\)](#)
- [Habilitação do multi-AZ \(API do ElastiCache\)](#)

Habilitação do multi-AZ (console)

É possível habilitar o Multi-AZ usando o console do ElastiCache ao criar um novo cluster do Valkey ou Redis OSS ou modificando um cluster existente com replicação.

O Multi-AZ é habilitado por padrão em clusters do Valkey ou Redis OSS (modo cluster habilitado).

 Important

O ElastiCache habilitará automaticamente o multi-AZ somente se o cluster contiver pelo menos uma réplica em uma zona de disponibilidade diferente da principal em todos os fragmentos.

Habilitação do multi-AZ ao criar um cluster usando o console do ElastiCache

Para obter mais informações sobre esse processo, consulte [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#). Tenha uma ou mais réplicas e habilite o Multi-AZ.

Habilitação do multi-AZ em um cluster existente (console)

Para obter mais informações sobre esse processo, consulte Modificação de um cluster [Uso do Console de gerenciamento da AWS do ElastiCache](#).

Habilitar o recurso multi-AZ (AWS CLI)

O exemplo de código a seguir usa a AWS CLI para habilitar o Multi-AZ para o grupo de replicação `redis12`.

 Important

O grupo de replicação `redis12` já deve existir e ter pelo menos uma réplica de leitura disponível.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id redis12 \  
  --automatic-failover-enabled \  
  --multi-az-enabled \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id redis12 ^
```

```
--automatic-failover-enabled ^  
--multi-az-enabled ^  
--apply-immediately
```

A saída JSON desse comando deve ser semelhante ao seguinte.

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "One shard, two nodes",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2b",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis12-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis12-002"  
          }  
        ],  
        "NodeGroupId": "0001",  
        "PrimaryEndpoint": {  
          "Port": 6379,  
          "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"  
        }  
      }  
    ]  
  }  
}
```

```
    ],  
    "ReplicationGroupId": "redis12",  
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "enabling",  
    "MultiAZ": "enabled",  
    "SnapshotWindow": "07:00-08:00",  
    "SnapshottingClusterId": "redis12-002",  
    "MemberClusters": [  
        "redis12-001",  
        "redis12-002"  
    ],  
    "PendingModifiedValues": {}  
  }  
}
```

Para obter mais informações, consulte estes tópicos na AWS CLI Referência de comandos:

- [create-cache-cluster](#)
- [create-replication-group](#)
- [modify-replication-group](#) na AWS CLI Referência de comandos.

Habilitação do multi-AZ (API do ElastiCache)

O exemplo de código a seguir usa a API do ElastiCache para habilitar o multi-AZ para o grupo de replicação `redis12`.

Note

Para usar esse exemplo, o grupo de replicação `redis12` já deve existir e ter pelo menos uma réplica de leitura disponível.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ApplyImmediately=true  
&AutoFailover=true  
&MultiAZEnabled=true  
&ReplicationGroupId=redis12  
&Version=2015-02-02  
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256  
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, veja estes tópicos na Referência da API do ElastiCache:

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Cenários de falha com respostas do multi-AZ

Antes da introdução do multi-AZ, o ElastiCache detectava e substituíam os nós com falha de um cluster recriando e reprovisionando esses nós. Se você habilitar o Multi-AZ, um nó primário com falha fará failover para a réplica com o menor atraso de replicação. A réplica selecionada é promovida automaticamente para primário, o que é muito mais rápido do que criar e reprovisionar um novo nó primário. Esse processo normalmente demora apenas alguns segundos até que você possa gravar novamente no cluster.

Quando o multi-AZ está habilitado, o ElastiCache monitora continuamente o estado do nó primário. Se o nó primário falhar, uma das seguintes ações será realizada, dependendo do tipo da falha.

Tópicos

- [Cenários de falha quando somente o nó primário falha](#)
- [Cenários de falha quando o nó primário e algumas réplicas de leitura falham](#)
- [Cenários de falha quando cluster inteiro falha](#)

Cenários de falha quando somente o nó primário falha

Se somente o nó primário falhar, a réplica de leitura com o menor atraso de replicação será promovida a primária. Depois disso, uma réplica de leitura de substituição é criada e provisionada na mesma zona de disponibilidade que o primário com falha.

Quando somente o nó primário falha, o recurso multi-AZ do ElastiCache faz o seguinte:

1. O nó primário com falha é colocado offline.
2. A réplica de leitura com o menor atraso de replicação é promovida a primário.

As gravações poderão ser retomadas assim que o processo de promoção estiver concluído, normalmente depois de apenas alguns segundos. Se a sua aplicação estiver gravando no endpoint primário, não será necessário alterar o endpoint para gravações ou leituras. O ElastiCache propaga o nome do DNS da réplica promovida.

3. Uma réplica de leitura de substituição é executada e provisionada.

A réplica de leitura de substituição é executada na Zona de disponibilidade em que o nó primário com falha se encontrava, para que a distribuição de nós seja mantida.

4. As réplicas são sincronizadas com o novo nó primário.

Depois que a nova réplica estiver disponível, lembre-se dos seguintes efeitos:

- Endpoint primário: não faça alterações na sua aplicação, pois o nome do DNS do novo nó primário é propagado para o endpoint primário.
- Endpoint leitor: o endpoint leitor é atualizado automaticamente para apontar para os novos nós de réplica.

Para obter informações sobre como encontrar os endpoints de um cluster, consulte os seguintes tópicos:

- [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(API do ElastiCache\)](#)

Cenários de falha quando o nó primário e algumas réplicas de leitura falham

Se o primário e pelo menos uma réplica de leitura falhar, a réplica disponível com o menor atraso de replicação será promovida a cluster primário. Novas réplicas de leitura também são criadas e provisionadas nas mesmas Zonas de disponibilidade que os nós com falha e a réplica que foi promovida a primário.

Quando o nó primário e algumas réplicas de leitura falham, o multi-AZ do ElastiCache faz o seguinte:

1. O nó primário com falha e as réplicas de leitura com falha são colocadas offline.
2. A réplica disponível com o menor atraso de replicação é promovida a nó primário.

As gravações poderão ser retomadas assim que o processo de promoção estiver concluído, normalmente depois de apenas alguns segundos. Se a sua aplicação estiver gravando no endpoint primário, não será necessário alterar o endpoint para gravações. O ElastiCache propaga o nome do DNS da réplica promovida.

3. Réplicas de substituição são criadas e provisionadas.

As réplicas de substituição são criadas nas Zonas de disponibilidade dos nós com falha, de modo que a distribuição de nós seja mantida.

4. Todos os clusters são sincronizados com o novo nó primário.

Faça as seguintes alterações no seu aplicativo depois que os novos nós estiverem disponíveis:

- Endpoint primário: não faça alterações em sua aplicação. O nome de DNS do novo nó primário é propagado para o endpoint primário.
- Endpoint leitor: o endpoint leitor é atualizado automaticamente para apontar para os novos nós de réplica.

Para obter informações sobre como encontrar os endpoints de um grupo de replicação, consulte os seguintes tópicos:

- [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(API do ElastiCache\)](#)

Cenários de falha quando cluster inteiro falha

Se tudo falhar, todos os nós serão recriados e provisionados nas mesmas Zonas de disponibilidade que os nós originais.

Nesse cenário, todos os dados do cluster são perdidos devido à falha de cada nó no cluster. Essa ocorrência é rara.

Quando o cluster inteiro falha, o multi-AZ do ElastiCache faz o seguinte:

1. O nó primário e as réplicas de leitura com falha são colocados offline.
2. Um nó primário de substituição é criado e provisionado.
3. Réplicas de substituição são criadas e provisionadas.

As substituições são criadas nas Zonas de disponibilidade dos nós com falha, de modo que a distribuição de nós seja mantida.

Como o cluster inteiro falhou, os dados são perdidos, e todos os novos nós são iniciados a frio.

Como cada um dos nós de substituição tem o mesmo endpoint que o nó que ele está substituindo, não é necessário fazer alterações de endpoint no seu aplicativo.

Para obter informações sobre como encontrar os endpoints de um grupo de replicação, consulte os seguintes tópicos:

- [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(API do ElastiCache\)](#)

Recomendamos que você crie o nó primário e as réplicas de leitura em diferentes Zonas de disponibilidade para aumentar o nível de tolerância a falhas.

Teste do failover automático

Depois de habilitar o failover automático, é possível testá-lo usando o console do ElastiCache, a AWS CLI e a API do ElastiCache.

Ao testar, observe o seguinte:

- Você pode usar essa operação para testar o failover automático em até 15 fragmentos (chamados de grupos de nós na API do ElastiCache e na AWS CLI) em qualquer período contínuo de 24 horas.
- Se você chamar essa operação em estilhaços em clusters diferentes (chamados grupos de replicação na API e CLI), poderá fazer as chamadas simultaneamente.
- Em alguns casos, é possível chamar essa operação várias vezes em diferentes fragmentos no mesmo grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Nesses casos, a substituição do primeiro nó deve ser concluída antes que uma chamada subsequente possa ser feita.
- Para determinar se a substituição do nó está completa, verifique os eventos usando o console do Amazon ElastiCache, a AWS CLI ou a API do ElastiCache. Procure pelos seguintes eventos relacionados ao failover automático, listados aqui em ordem de ocorrência:
 1. Mensagem do grupo de replicação: `Test Failover API called for node group <node-group-id>`
 2. Mensagem do cluster de cache: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 3. Mensagem do grupo de replicação: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 4. Mensagem do cluster de cache: `Recovering cache nodes <node-id>`
 5. Mensagem do cluster de cache: `Finished recovery for cache nodes <node-id>`

Para obter mais informações, consulte:

- [Visualizando ElastiCache eventos](#) no Guia do usuário do ElastiCache
- [DescribeEvents](#) na Referência da API do ElastiCache
- [describe-events](#) na AWS CLI Referência de comandos.
- Essa API foi projetada para testar o comportamento de sua aplicação em caso de failover do ElastiCache. Ela não foi projetado para ser uma ferramenta operacional para iniciar um failover a fim de resolver um problema com o cluster. Além disso, sob determinadas condições, como eventos operacionais de grande escala, a AWS pode bloquear essa API.

Tópicos

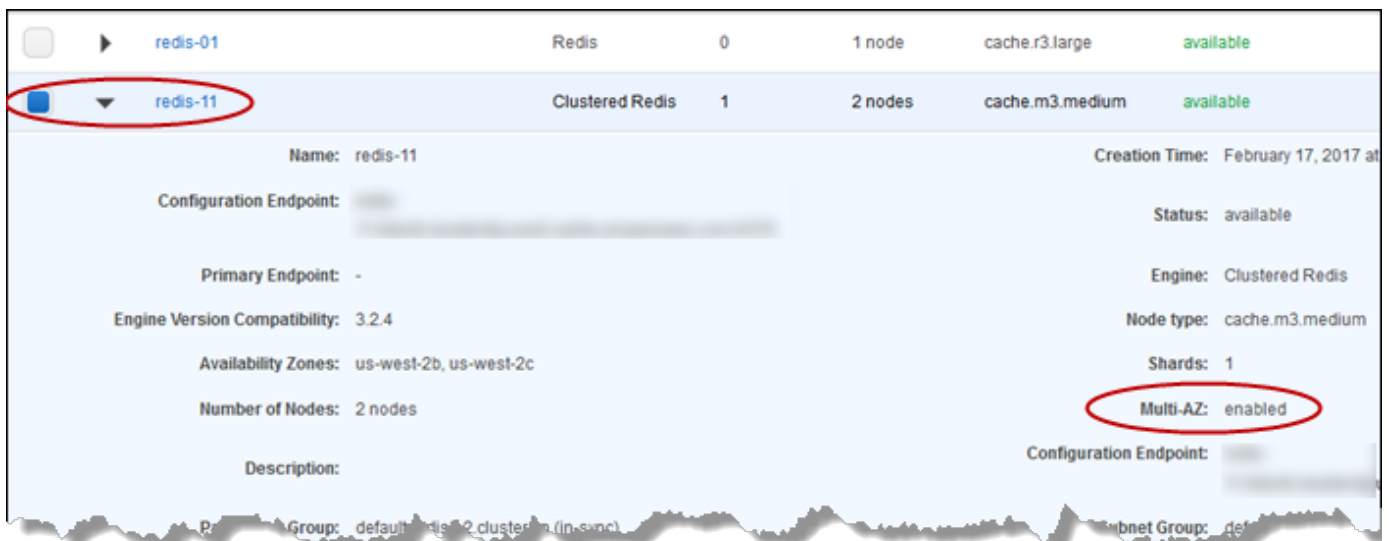
- [Teste do failover automático usando o Console de gerenciamento da AWS](#)
- [Teste do failover automático usando o AWS CLI](#)
- [Teste do failover automático usando a API do ElastiCache](#)

Teste do failover automático usando o Console de gerenciamento da AWS

Use o procedimento a seguir para testar o failover automático com o console.

Para testar o failover automático

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS.
3. Na lista de clusters, escolha a caixa à esquerda do cluster que deseja testar. Esse cluster deve ter pelo menos um nó de réplica de leitura.
4. Na área Details, confirme se esse cluster está habilitado para Multi-AZ. Se o cluster não estiver habilitado para o Multi-AZ, escolha um cluster diferente ou modifique esse cluster para habilitar o Multi-AZ. Para obter mais informações, consulte [Uso do Console de gerenciamento da AWS do ElastiCache](#).



5. Para o Valkey ou Redis OSS (modo cluster desabilitado), escolha o nome do cluster.

Para o Valkey ou Redis OSS (modo cluster habilitado), faça o seguinte:

- a. Escolha o nome do cluster.

- b. Na página Shards, para o estilhaço (chamado de grupo de nó na API e na CLI) no qual você deseja testar o failover, escolha o nome do estilhaço.
6. Na página Nodes, escolha Failover Primary.
7. Escolha Continue para fazer failover do primário ou Cancel para cancelar a operação e não fazer failover do nó primário.

Durante o processo de failover, o console continua a mostrar o status do nó como disponível. Para acompanhar o progresso do seu teste de failover, escolha Events no painel de navegação do console. Na guia Eventos, observe os eventos que indicam que o failover foi iniciado (Test Failover API called) e concluído (Recovery completed).

Teste do failover automático usando o AWS CLI

É possível testar o failover automático em qualquer cluster habilitado para Multi-AZ usando a operação da AWS CLI `test-failover`.

Parâmetros

- `--replication-group-id`: obrigatório. O grupo de replicação (no console, cluster) que deve ser testado.
- `--node-group-id`: obrigatório. O nome do grupo de nós nos quais você deseja testar o failover automático. Você pode testar um máximo de 15 grupos de nós em um período contínuo de 24 horas.

O exemplo a seguir usa a AWS CLI para testar o failover automático no grupo de nós `redis00-0003` do cluster do Valkey ou Redis OSS (modo cluster habilitado) `redis00`.

Exemplo Teste de failover automático

Para Linux, macOS ou Unix:

```
aws elasticache test-failover \  
  --replication-group-id redis00 \  
  --node-group-id redis00-0003
```

Para Windows:

```
aws elasticache test-failover ^  
--replication-group-id redis00 ^  
--node-group-id redis00-0003
```

A saída do comando precedente é semelhante ao seguinte.

```
{  
  "ReplicationGroup": {  
    "Status": "available",  
    "Description": "1 shard, 3 nodes (1 + 2 replicas)",  
    "NodeGroups": [  
      {  
        "Status": "available",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2c",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-002"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2b",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,
```



```

        "Address":
"redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"
    },
    "CacheClusterId": "redis1x3-003"
  }
],
"NodeGroupId": "0001",
"PrimaryEndpoint": {
  "Port": 6379,
  "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
}
}
],
"ClusterEnabled": false,
"ReplicationGroupId": "redis1x3",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotWindow": "11:30-12:30",
"SnapshottingClusterId": "redis1x3-002",
"MemberClusters": [
  "redis1x3-001",
  "redis1x3-002",
  "redis1x3-003"
],
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled",
"PendingModifiedValues": {}
}
}

```

Para acompanhar o progresso do failover, use a operação da AWS CLI `describe-events`.

Para obter mais informações, consulte:

- [test-failover](#) na AWS CLI Referência de comandos.
- [describe-events](#) na AWS CLI Referência de comandos.

Teste do failover automático usando a API do ElastiCache

É possível testar o failover automático em qualquer cluster habilitado com multi-AZ usando a operação de API do ElastiCache `TestFailover`.

Parâmetros

- `ReplicationGroupId`: obrigatório. O grupo de replicação (no console ou cluster) a ser testado.
- `NodeGroupId`: obrigatório. O nome do grupo de nós nos quais você deseja testar o failover automático. Você pode testar um máximo de 15 grupos de nós em um período contínuo de 24 horas.

O exemplo a seguir testa o failover automático no grupo de nós `redis00-0003` no grupo de replicação (no console, cluster) `redis00`.

Exemplo Teste do failover automático

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=TestFailover  
&NodeGroupId=redis00-0003  
&ReplicationGroupId=redis00  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Para acompanhar o progresso do failover, use a operação de API do ElastiCache `DescribeEvents`.

Para obter mais informações, consulte:

- [TestFailover](#) na Referência da API do ElastiCache
- [DescribeEvents](#) na Referência da API do ElastiCache

Limitações no Multi-AZ

Esteja ciente das seguintes limitações para o Multi-AZ:

- O Multi-AZ é compatível com Valkey e com Redis OSS versão 2.8.6 e posteriores.
- O Multi-AZ não é aceito em tipos de nós T1.
- A replicação do Valkey e Redis OSS é assíncrona. Portanto, quando um nó primário faz failover em uma réplica, uma pequena quantidade de dados pode ser perdida devido ao atraso da replicação.

Ao escolher a réplica a ser promovida a primária, o ElastiCache escolhe a réplica com o menor atraso de replicação. Em outras palavras, ele escolhe a réplica que é mais atual. Isso ajuda a minimizar a quantidade de dados perdidos. A réplica com o atraso de replicação mínimo pode estar na mesma zona de disponibilidade ou em outra em relação ao nó primário com falha.

- Ao promover manualmente réplicas de leitura para primárias em clusters do Valkey ou Redis OSS com o modo cluster desabilitado, você só poderá fazer isso quando o Multi-AZ e o failover automático estiverem desabilitados. Para promover uma réplica de leitura para primário, execute as seguintes etapas:
 1. Desabilite o Multi-AZ no cluster.
 2. Desabilite o failover automático no cluster. Isso pode ser feito por meio do console desmarcando a caixa de seleção Failover automático para o grupo de replicação. Também é possível fazer isso usando a AWS CLI definindo a propriedade `AutomaticFailoverEnabled` como `false` ao chamar a operação `ModifyReplicationGroup`.
 3. Promova a réplica de leitura para primário.
 4. Habilite novamente o Multi-AZ.
- O multi-AZ do ElastiCache for Redis OSS e o arquivo somente de acréscimo (AOF) são mutuamente exclusivos. Se você habilitar um, não é possível habilitar o outro.
- Uma falha de um nó pode ser causada pelo evento raro de falha total de uma zona de disponibilidade. Neste caso, a réplica que substitui o primário com falha é criada somente quando a zona de disponibilidade volta a ficar ativa. Por exemplo, considere um grupo de replicação com o primário em AZ-a e réplicas em AZ-b e AZ-c. Se o primário falhar, a réplica com o menor atraso de replicação será promovida a cluster primário. O ElastiCache criará uma nova réplica em AZ-a (onde o primário com falha estava localizado) somente quando AZ-a voltar a ficar ativa e disponível.
- Uma reinicialização iniciada pelo cliente de um primário não aciona o failover automático. Outras reinicializações e falhas desencadeiam o failover automático.
- Quando o primário é reiniciado, seus dados são limpos quando ele volta a ficar online. Quando as réplicas de leitura veem o cluster primário limpo, elas limpam suas cópias dos dados, o que causa perda de dados.

- Depois que uma réplica de leitura foi promovida, as outras réplicas se sincronizam com o novo primário. Após a sincronização inicial, o conteúdo das réplicas é excluído, e eles sincronizam os dados do novo primário. Esse processo de sincronização causa uma breve interrupção, durante a qual as réplicas não são acessíveis. O processo de sincronização também causa um aumento de carga temporário no primário durante a sincronização com as réplicas. Esse comportamento é nativo do Valkey e do Redis OSS e não é exclusivo do Multi-AZ do ElastiCache. Para obter detalhes sobre esse comportamento, consulte [Replicação](#) no site do Valkey.

 Important

Para Valkey 7.2.6 e posterior ou Redis OSS versão 2.8.22 e posterior, não é possível criar réplicas externas.

Para versões do Redis OSS anteriores à 2.8.22, recomendamos que você não conecte uma réplica externa a um cluster do ElastiCache que esteja habilitado para Multi-AZ. Essa configuração sem suporte pode criar problemas que impedem que o ElastiCache realize corretamente o processo de failover e recuperação. Para conectar uma réplica externa a um cluster do ElastiCache, verifique se o multi-AZ não está habilitado antes de fazer a conexão.

Como a sincronização e o backup são implementados

Todas as versões com suporte do Valkey ou Redis OSS oferecem suporte para backup e sincronização entre os nós primário e de réplica. No entanto, a forma como o backup e a sincronização são implementados varia de acordo com a versão.

Redis OSS versão 2.8.22 e posterior

Para a replicação Redis OSS, nas versões 2.8.22 e posteriores, escolha um dos dois métodos de backup. Para obter mais informações, consulte [Versões do Redis OSS anteriores a 2.8.22](#) e [Snapshots e restauração](#).

Durante o processo sem permissão, se as cargas de gravação forem pesadas, as gravações no cluster serão atrasadas para garantir que você não acumule muitas mudanças e, assim, evite um snapshot bem-sucedido.

Versões do Redis OSS anteriores a 2.8.22

O backup e a sincronização do Redis OSS nas versões anteriores a 2.8.22 é um processo de três etapas.

1. A bifurcação e, no processo em segundo plano, serializam os dados do cluster no disco. Isso cria um snapshot de ponto no tempo.
2. Em primeiro plano, acumule um log de alterações no buffer de saída do cliente.

Important

Se o log de alterações exceder o tamanho do buffer de saída do cliente, o backup ou a sincronização falhará. Para obter mais informações, consulte [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#).

3. Por fim, transmita os dados do cache e, em seguida, o log de alterações para o nó de réplica.

Criação de um grupo de replicação do Valkey ou Redis OSS

Você tem as seguintes opções para criar um cluster com nós de réplica. Uma delas se aplica quando você já tem um cluster do Valkey ou Redis OSS (modo cluster desabilitado) disponível não associado a nenhum cluster que possua réplicas a serem usadas como o nó primário. A outra é aplicada quando você precisa criar um nó primário com o cluster e as réplicas de leitura. Atualmente, um cluster do Valkey ou Redis OSS (modo cluster habilitado) deve ser criado do zero.

Opção 1 1: [Criação de um grupo de replicação usando um cluster existente](#)

Use essa opção para aproveitar um cluster de nó único já existente do Valkey ou Redis OSS (modo cluster desabilitado). Você especifica esse nó existente como o nó primário no novo cluster e, em seguida, adiciona individualmente de 1 a 5 réplicas de leitura ao cluster. Se o cluster existente estiver ativo, as réplicas de leitura serão sincronizadas com ele enquanto forem criadas. Consulte [Criação de um grupo de replicação usando um cluster existente](#).

Important

Não é possível criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) usando um cluster existente. Para criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) (API/CLI: grupo de replicação) usando o console do ElastiCache, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#).

Opção 2 2: [Criação de um grupo de replicação do Valkey ou Redis OSS do início](#)

Use essa opção se você ainda não tiver um cluster do Valkey ou Redis OSS (modo cluster desabilitado) disponível a ser usado como nó primário do cluster, ou se quiser criar um cluster do Valkey ou Redis OSS (modo cluster habilitado). Consulte [Criação de um grupo de replicação do Valkey ou Redis OSS do início](#).

Criação de um grupo de replicação usando um cluster existente

O procedimento a seguir adiciona um grupo de replicação ao cluster de nó único do Valkey ou Redis OSS (modo cluster desabilitado). Ele é necessário para atualizar seu cluster para a versão mais recente do Valkey. Esse é um procedimento local que envolve zero tempo de inatividade e zero perda de dados. Quando você cria um grupo de replicação para seu cluster de nó único, o nó do cluster se torna o nó primário no novo cluster. Se você não tiver um cluster do Valkey ou Redis OSS (modo cluster desabilitado) que possa usar como primário do novo cluster, consulte [Criação de um grupo de replicação do Valkey ou Redis OSS do início](#).

Um cluster disponível é um cluster do Valkey ou Redis OSS existente de um único nó. Atualmente, o Valkey ou o Redis OSS (modo cluster habilitado) não oferecem suporte para criar um cluster com réplicas usando um cluster de um único nó disponível. Se você quiser criar um cluster do Valkey ou Redis OSS (modo cluster habilitado), consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#).

Criação de um grupo de replicação usando um cluster existente (console)

Consulte o tópico [Usando o ElastiCache Console de gerenciamento da AWS](#).

Criação de um grupo de replicação usando um cluster do Valkey ou Redis OSS disponível (AWS CLI)

Existem duas etapas para criar um grupo de replicação com réplicas de leitura ao usar um cluster de cache do Valkey ou Redis OSS disponível para o primário ao usar a AWS CLI.

Ao usar o AWS CLI você cria um grupo de replicação especificando o nó autônomo disponível como o nó principal do cluster `--primary-cluster-id` e o número de nós que você deseja no cluster usando o comando CLI, `create-replication-group`. Inclua os seguintes parâmetros.

`--replication-group-id`

O nome do grupo de replicação que você está criando. O valor desse parâmetro é usado como a base para os nomes dos nós adicionados com um número sequencial de 3 dígitos acrescentados ao final de `--replication-group-id`. Por exemplo, `sample-repl-group-001`.

As restrições de nomenclatura do grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.

- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`--replication-group-description`

Descrição do grupo de replicação.

`--num-node-groups`

O número de nós desejados neste cluster. Esse valor inclui o nó primário. Esse parâmetro tem um valor máximo de seis.

`--primary-cluster-id`

O nome do cluster do Valkey ou Redis OSS (modo cluster desabilitado) disponível que você deseja que seja o nó primário nesse grupo de replicação.

O comando a seguir cria o grupo de replicação `sample-repl-group` usando o cluster do Valkey ou Redis OSS (modo cluster desabilitado) disponível `redis01` como o nó primário do grupo de replicação. Ele cria 2 novos nós que são réplicas de leitura. As configurações de `redis01` (ou seja, grupo de parâmetros, grupo de segurança, tipo de nó, versão do mecanismo, etc.) serão aplicadas a todos os nós no grupo de replicações.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --primary-cluster-id redis01
```

Para obter informações e parâmetros adicionais que você talvez queira usar, consulte o AWS CLI tópico [create-replication-group](#).

Em seguida, adicione réplicas de leitura ao grupo de replicação

Após a criação do grupo de replicação, adicione de uma a cinco réplicas de leitura usando o comando `create-cache-cluster`, certificando-se de incluir os seguintes parâmetros.

`--cache-cluster-id`

O nome do cluster que você está adicionando ao grupo de replicação.

As restrições de nomenclatura de cluster são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`--replication-group-id`

O nome do grupo de replicação ao qual você está adicionando esse cluster.

Repita esse comando para cada réplica de leitura que você deseja adicionar ao grupo de replicação, alterando apenas o valor do parâmetro `--cache-cluster-id`.

Note

Lembre-se de que um grupo de replicação não pode ter mais de cinco réplicas de leitura. Se você tentar adicionar uma réplica de leitura a um grupo de replicação que já possui cinco réplicas de leitura, a operação falhará.

O código a seguir adiciona a réplica de leitura `my-replica01` ao grupo de replicação `sample-repl-group`. As configurações do cluster primário (grupo de parâmetros, grupo de segurança, tipo de nó etc.) serão aplicadas aos nós como foram adicionadas ao grupo de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-replica01 \  
  --replication-group-id sample-repl-group
```

Para Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-replica01 ^  
  --replication-group-id sample-repl-group
```

A saída desse comando será semelhante a esta.

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "demo cluster with replicas",  
    "ClusterEnabled": false,  
    "ReplicationGroupId": "sample-repl-group",  
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "disabled",  
    "SnapshotWindow": "00:00-01:00",  
    "SnapshottingClusterId": "redis01",  
    "MemberClusters": [  
      "sample-repl-group-001",  
      "sample-repl-group-002",  
      "redis01"  
    ],  
    "CacheNodeType": "cache.m4.large",  
    "DataTiering": "disabled",  
    "PendingModifiedValues": {}  
  }  
}
```

Para obter informações adicionais, consulte os AWS CLI tópicos:

- [create-replication-group](#)
- [modify-replication-group](#)

Adicionar réplicas a um cluster (API) autônomo Valkey ou Redis OSS (modo de cluster desativado)
ElastiCache

Ao usar a ElastiCache API, você cria um grupo de replicação especificando o nó autônomo disponível como o nó principal do cluster `PrimaryClusterId` e o número de nós que você deseja no cluster usando o comando CLI. `CreateReplicationGroup` Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando. O valor desse parâmetro é usado como a base para os nomes dos nós adicionados com um número sequencial de 3 dígitos acrescentados ao final de `ReplicationGroupId`. Por exemplo, `.sample-repl-group-001`

As restrições de nomenclatura do grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

ReplicationGroupDescription

Descrição do cluster com réplicas.

NumCacheClusters

O número de nós desejados neste cluster. Esse valor inclui o nó primário. Esse parâmetro tem um valor máximo de seis.

PrimaryClusterId

O nome do cluster do Valkey ou Redis OSS (modo cluster desabilitado) disponível que você deseja que seja o nó primário nesse cluster.

O comando a seguir cria o cluster com réplicas `sample-repl-group` usando o cluster do Valkey ou Redis OSS (modo cluster desabilitado) disponível `redis01` como o nó primário do grupo de replicação. Ele cria 2 novos nós que são réplicas de leitura. As configurações de `redis01` (ou seja, grupo de parâmetros, grupo de segurança, tipo de nó, versão do mecanismo, etc.) serão aplicadas a todos os nós no grupo de replicações.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateReplicationGroup
&Engine=redis
&EngineVersion=6.0
&ReplicationGroupDescription=Demo%20cluster%20with%20replicas
&ReplicationGroupId=sample-repl-group
&PrimaryClusterId=redis01
&Version=2015-02-02
```

```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter informações adicionais, consulte os tópicos do ElastiCache APL:

- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Em seguida, adicione réplicas de leitura ao grupo de replicação

Após a criação do grupo de replicação, adicione de uma a cinco réplicas de leitura usando a operação `CreateCacheCluster`, certificando-se de incluir os seguintes parâmetros.

`CacheClusterId`

O nome do cluster que você está adicionando ao grupo de replicação.

As restrições de nomenclatura de cluster são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`ReplicationGroupId`

O nome do grupo de replicação ao qual você está adicionando esse cluster.

Repita essa operação para cada réplica de leitura que você deseja adicionar ao grupo de replicação, alterando apenas o valor do parâmetro `CacheClusterId`.

O código a seguir adiciona a réplica de leitura `myReplica01` ao grupo de replicação `myReplGroup`. As configurações do cluster primário (grupo de parâmetros, grupo de segurança, tipo de nó etc.) serão aplicadas a nós conforme eles forem adicionados ao grupo de replicação.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheCluster
&CacheClusterId=myReplica01
```

```
&ReplicationGroupId=myReplGroup
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=[your-access-key-id]/20150202/us-west-2/elasticache/aws4_request
&X-Amz-Date=20150202T170651Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=[signature-value]
```

Para obter informações e parâmetros adicionais que você talvez queira usar, consulte o tópico [ElastiCache da API `CreateCacheCluster`](#).

Criação de um grupo de replicação do Valkey ou Redis OSS do início

Veja a seguir como criar um grupo de replicação do Valkey ou Redis OSS sem usar um cluster do Valkey ou Redis OSS existente como primário. Você pode criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) ou do Valkey ou Redis OSS (modo cluster habilitado) do início usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Antes de continuar, decida se deseja criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) ou do Valkey ou Redis OSS (modo cluster habilitado). Para obter orientação nessa decisão, consulte [Replicação: Modo de cluster Valkey e Redis OSS desativado versus ativado](#).

Tópicos

- [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) do início](#)
- [Criação de um grupo de replicação no Valkey ou Redis OSS \(modo cluster habilitado\) do início](#)

Criação de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) do início

Você pode criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) a partir do início usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) sempre possui um grupo de nó, um cluster primário e até cinco réplicas de leitura. Os grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado) não são compatíveis com o particionamento dos seus dados.

Note

O limite de nó/fragmento pode ser aumentado para um máximo de 500 por cluster. Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e inclua o tipo de instância na solicitação.

Para criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) do início, siga uma das seguintes abordagens:

Criação de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) do início (AWS CLI)

O procedimento a seguir cria um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) usando a AWS CLI.

Ao criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) do início, você cria o grupo de replicação e todos os seus nós com uma única chamada para o comando `create-replication-group` da AWS CLI. Inclua os seguintes parâmetros.

`--replication-group-id`

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura do grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`--replication-group-description`

Descrição do grupo de replicação.

`--num-cache-clusters`

O número de nós que você deseja criar com esse grupo de replicação, com réplicas primárias e de leitura combinadas.

Se você habilitar o Multi-AZ (`--automatic-failover-enabled`), o valor de `--num-cache-clusters` deverá ser pelo menos 2.

`--cache-node-type`

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das geração anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

`--data-tiering-enabled`

Defina esse parâmetro se estiver usando um tipo de nó `r6gd`. Se não quiser classificação de dados em níveis, defina `--no-data-tiering-enabled`. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

`--cache-parameter-group`

Especifique um grupo de parâmetros que corresponda à versão do seu mecanismo. Se estiver executando o Redis OSS 3.2.4 ou posterior, especifique o grupo de parâmetros `default.redis3.2` ou um grupo de parâmetros de `default.redis3.2` para criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado). Para obter mais informações, consulte [Parâmetros do Valkey e do Redis OSS](#).

`--network-type`

`ipv4`, `ipv6` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro `--IpDiscovery` como `ipv4` ou `ipv6`.

`--engine`

`redis`

--engine-version

Para ter o conjunto mais amplo de recursos, escolha a versão mais recente do mecanismo.

Os nomes dos nós serão derivados do nome do grupo de replicação depois de adicionar -00# ao nome do grupo de replicação. Por exemplo, usando o nome do grupo de replicação myReplGroup, o nome para o primário será myReplGroup-001 e as réplicas de leitura myReplGroup-002 e myReplGroup-006.

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros --transit-encryption-enabled ou --at-rest-encryption-enabled e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis OSS versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro --cache-subnet-group.
- Você também deve incluir o parâmetro --auth-token com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

A seguinte operação cria um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) sample-repl-group com três nós, um primário e duas réplicas.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --cache-node-type cache.m4.large \  
  --engine redis
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --cache-node-type cache.m4.large ^  
  --engine redis
```

A saída desse comando é semelhante a esta.

```
{
  "ReplicationGroup": {
    "Status": "creating",
    "Description": "Demo cluster with replicas",
    "ClusterEnabled": false,
    "ReplicationGroupId": "sample-repl-group",
    "SnapshotRetentionLimit": 0,
    "AutomaticFailover": "disabled",
    "SnapshotWindow": "01:30-02:30",
    "MemberClusters": [
      "sample-repl-group-001",
      "sample-repl-group-002",
      "sample-repl-group-003"
    ],
    "CacheNodeType": "cache.m4.large",
    "DataTiering": "disabled",
    "PendingModifiedValues": {}
  }
}
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico da AWS CLI [create-replication-group](#).

Criação de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) do início (API do ElastiCache)

O procedimento a seguir cria um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) usando a API do ElastiCache.

Ao criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) do início, você cria o grupo de replicação e todos os seus nós com uma única chamada para a operação `CreateReplicationGroup` da API do ElastiCache. Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura do grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.

- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

ReplicationGroupDescription

Sua descrição do grupo de replicação.

NumCacheClusters

O número total de nós que você deseja criar com esse grupo de replicação, com réplicas primárias e de leitura combinadas.

Se você habilitar o Multi-AZ (`AutomaticFailoverEnabled=true`), o valor de `NumCacheClusters` deverá ser pelo menos 2.

CacheNodeType

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das geração anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

--data-tiering-enabled

Defina esse parâmetro se estiver usando um tipo de nó `r6gd`. Se não quiser classificação de dados em níveis, defina `--no-data-tiering-enabled`. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

CacheParameterGroup

Especifique um grupo de parâmetros que corresponda à versão do seu mecanismo. Se estiver executando o Redis OSS 3.2.4 ou posterior, especifique o grupo de parâmetros `default.redis3.2` ou um grupo de parâmetros de `default.redis3.2` para criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado). Para obter mais informações, consulte [Parâmetros do Valkey e do Redis OSS](#).

--network-type

`ipv4`, `ipv` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro `--IpDiscovery` como `ipv4` ou `ipv6`.

Mecanismo

redis

EngineVersion

6.0

Os nomes dos nós serão derivados do nome do grupo de replicação depois de adicionar -00# ao nome do grupo de replicação. Por exemplo, usando o nome do grupo de replicação myReplGroup, o nome para o primário será myReplGroup-001 e as réplicas de leitura myReplGroup-002 e myReplGroup-006.

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros TransitEncryptionEnabled=true ou AtRestEncryptionEnabled=true e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis OSS versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro CacheSubnetGroup.
- Você também deve incluir o parâmetro AuthToken com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

A seguinte operação cria o grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) myReplGroup com três nós, um primário e duas réplicas.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&CacheNodeType=cache.m4.large  
&CacheParameterGroup=default.redis6.x  
&Engine=redis  
&EngineVersion=6.0  
&NumCacheClusters=3  
&ReplicationGroupDescription=test%20group  
&ReplicationGroupId=myReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a API do ElastiCache [CreateReplicationGroup](#).

Criação de um grupo de replicação no Valkey ou Redis OSS (modo cluster habilitado) do início

Você pode criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) (API/CLI: grupo de replicação) usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) tem de 1 a 500 fragmentos (API/CLI: grupos de nós), um nó primário em cada fragmento e até 5 réplicas de leitura em cada fragmento. É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo Valkey ou Redis OSS for 5.0.6 ou posterior. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Criação de um cluster no Valkey ou Redis OSS (modo cluster habilitado)

- [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)
- [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster habilitado\) do início \(AWS CLI\)](#)
- [Criação de um grupo de replicação no Valkey ou Redis OSS \(modo cluster habilitado\) do início \(API do ElastiCache\)](#)

Criação de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (console)

Para criar um cluster do Valkey ou Redis OSS (modo cluster habilitado), consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#). Certifique-se de habilitar o modo de cluster, Cluster Mode enabled (Scale Out), e especifique pelo menos dois fragmentos e um nó de réplica em cada.

Criação de um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) do início (AWS CLI)

O procedimento a seguir cria um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) usando a AWS CLI.

Ao criar um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) do início, você cria o grupo de replicação e todos os seus nós com uma única chamada para o comando `create-replication-group` da AWS CLI. Inclua os seguintes parâmetros.

`--replication-group-id`

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura do grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`--replication-group-description`

Descrição do grupo de replicação.

`--cache-node-type`

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das gerações anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

`--data-tiering-enabled`

Defina esse parâmetro se estiver usando um tipo de nó `r6gd`. Se não quiser classificação de dados em níveis, defina `--no-data-tiering-enabled`. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

`--cache-parameter-group`

Especifique o grupo de parâmetros `default.redis6.x.cluster.on` ou um grupo de parâmetros derivado de `default.redis6.x.cluster.on` para criar um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Para obter mais informações, consulte [Alterações de parâmetros do Redis OSS 6.x](#).

`--engine`

`redis`

`--engine-version`

`3.2.4`

`--num-node-groups`

O número de grupos de nós neste grupo de replicação. Os valores válidos são 1 a 500.

Note

O limite de nó/fragmento pode ser aumentado para um máximo de 500 por cluster. Para solicitar um aumento do limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite "Nodes per cluster per instance type" (Nós por cluster por tipo de instância).

`--replicas-per-node-group`

O número de nós de réplica em cada grupo de nós. Os valores válidos são 0 a 5.

`--network-type`

`ipv4`, `ipv` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro `--IpDiscovery` como `ipv4` ou `ipv6`.

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros `--transit-encryption-enabled` ou `--at-rest-encryption-enabled` e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis OSS versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.

- Você também deve incluir o parâmetro `--cache-subnet-group`.
- Você também deve incluir o parâmetro `--auth-token` com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

A seguinte operação cria o grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) `sample-repl-group` com três grupos de nós/fragmentos (`--num-node-groups`), cada um com três nós, uma réplica primária e duas réplicas de leitura (`--replicas-per-node-group`).

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-node-groups 3 \  
  --replicas-per-node-group 2 \  
  --cache-node-type cache.m4.large \  
  --engine redis \  
  --security-group-ids SECURITY_GROUP_ID \  
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --cache-node-type cache.m4.large ^  
  --engine redis ^  
  --security-group-ids SECURITY_GROUP_ID ^  
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

O comando anterior gera a seguinte saída.

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "Demo cluster with replicas",  
    "ReplicationGroupId": "sample-repl-group",
```

```
    "SnapshotRetentionLimit": 0,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "05:30-06:30",
    "MemberClusters": [
        "sample-repl-group-0001-001",
        "sample-repl-group-0001-002",
        "sample-repl-group-0001-003",
        "sample-repl-group-0002-001",
        "sample-repl-group-0002-002",
        "sample-repl-group-0002-003",
        "sample-repl-group-0003-001",
        "sample-repl-group-0003-002",
        "sample-repl-group-0003-003"
    ],
    "PendingModifiedValues": {}
}
```

Quando você cria um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) do início, você pode configurar cada fragmento do cluster usando o parâmetro `--node-group-configuration`, como mostra o exemplo a seguir, que configura dois grupos de nós (console: fragmentos). O primeiro fragmento tem dois nós: um primário e uma réplica de leitura. O segundo fragmento tem três nós: um primário e duas réplicas de leitura.

`--node-group-configuration`

A configuração para cada grupo de nós. O parâmetro `--node-group-configuration` consiste nos seguintes campos.

- **PrimaryAvailabilityZone:** a zona de disponibilidade onde o nó primário desse grupo de nós está localizado. Se esse parâmetro for omitido, o ElastiCache escolherá a zona de disponibilidade para o nó primário.

Exemplo: `us-west-2a`.

- **ReplicaAvailabilityZones:** uma lista separada por vírgulas de zonas de disponibilidade em que as réplicas de leitura estão localizadas. O número de Zonas de disponibilidade nessa lista deve corresponder ao valor de `ReplicaCount`. Se esse parâmetro for omitido, o ElastiCache escolherá as zonas de disponibilidade para os nós de réplica.

Exemplo: `"us-west-2a,us-west-2b,us-west-2c"`

- **ReplicaCount:** o número de nós de réplica nesse grupo de nós.

- **Slots:** uma string que especifica o espaço de chaves para o grupo de nós. A string está no formato `startKey-endKey`. Se esse parâmetro for omitido, o ElastiCache alocará as chaves igualmente entre os grupos de nós.

Exemplo: "0-4999"

A operação a seguir cria o grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) `new-group` com dois grupos de nós/fragmentos (`--num-node-groups`). Ao contrário do exemplo anterior, cada grupo de nós está configurado de forma diferente do outro grupo de nós (`--node-group-configuration`).

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id new-group \  
  --replication-group-description "Sharded replication group" \  
  --engine redis \  
  --snapshot-retention-limit 8 \  
  --cache-node-type cache.m4.medium \  
  --num-node-groups 2 \  
  --node-group-configuration \  
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \  
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a', 'us-east-1c'"
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id new-group ^  
  --replication-group-description "Sharded replication group" ^  
  --engine redis ^  
  --snapshot-retention-limit 8 ^  
  --cache-node-type cache.m4.medium ^  
  --num-node-groups 2 ^  
  --node-group-configuration \  
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \  
    "
```

```
"ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

A operação anterior gera a seguinte saída.

```
{
  "ReplicationGroup": {
    "Status": "creating",
    "Description": "Sharded replication group",
    "ReplicationGroupId": "rc-rg",
    "SnapshotRetentionLimit": 8,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "10:00-11:00",
    "MemberClusters": [
      "rc-rg-0001-001",
      "rc-rg-0001-002",
      "rc-rg-0002-001",
      "rc-rg-0002-002",
      "rc-rg-0002-003"
    ],
    "PendingModifiedValues": {}
  }
}
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a AWS CLI [create-replication-group](#).

Criação de um grupo de replicação no Valkey ou Redis OSS (modo cluster habilitado) do início (API do ElastiCache)

O procedimento a seguir cria um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) usando a API do ElastiCache.

Ao criar um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) do início, você cria o grupo de replicação e todos os seus nós com uma única chamada para a operação `CreateReplicationGroup` da API do ElastiCache. Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura do grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

ReplicationGroupDescription

Descrição do grupo de replicação.

NumNodeGroups

O número de grupos de nós que você deseja criar com esse grupo de replicação. Os valores válidos são 1 a 500.

ReplicasPerNodeGroup

O número de nós de réplica em cada grupo de nós. Os valores válidos são 1 a 5.

NodeGroupConfiguration

A configuração para cada grupo de nós. O parâmetro `NodeGroupConfiguration` consiste nos seguintes campos.

- `PrimaryAvailabilityZone`: a zona de disponibilidade onde o nó primário desse grupo de nós está localizado. Se esse parâmetro for omitido, o ElastiCache escolherá a zona de disponibilidade para o nó primário.

Exemplo: `us-west-2a`.

- `ReplicaAvailabilityZones`: uma lista de zonas de disponibilidade em que as réplicas de leitura estão localizadas. O número de Zonas de disponibilidade nessa lista deve corresponder ao valor de `ReplicaCount`. Se esse parâmetro for omitido, o ElastiCache escolherá as zonas de disponibilidade para os nós de réplica.
- `ReplicaCount`: o número de nós de réplica nesse grupo de nós.
- `Slots`: uma string que especifica o espaço de chaves para o grupo de nós. A string está no formato `startKey-endKey`. Se esse parâmetro for omitido, o ElastiCache alocará as chaves igualmente entre os grupos de nós.

Exemplo: `"0-4999"`

CacheNodeType

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das gerações anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

--data-tiering-enabled

Defina esse parâmetro se estiver usando um tipo de nó r6gd. Se não quiser classificação de dados em níveis, defina --no-data-tiering-enabled. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

CacheParameterGroup

Especifique o grupo de parâmetros `default.redis6.x.cluster.on` ou um grupo de parâmetros derivado de `default.redis6.x.cluster.on` para criar um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Para obter mais informações, consulte [Alterações de parâmetros do Redis OSS 6.x](#).

--network-type

`ipv4`, `ipv` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro --IpDiscovery como `ipv4` ou `ipv6`.

Mecanismo

`redis`

EngineVersion

`6.0`

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros `TransitEncryptionEnabled=true` ou `AtRestEncryptionEnabled=true` e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis OSS versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro `CacheSubnetGroup`.

- Você também deve incluir o parâmetro AuthToken com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

As quebras de linha foram adicionadas para legibilidade.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateReplicationGroup  
  &CacheNodeType=cache.m4.large  
  &CacheParameterGroup=default.redis6.xcluster.on  
  &Engine=redis  
  &EngineVersion=6.0  
  &NumNodeGroups=3  
  &ReplicasPerNodeGroup=2  
  &ReplicationGroupDescription=test%20group  
  &ReplicationGroupId=myReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a API do ElastiCache [CreateReplicationGroup](#).

Visualização dos detalhes de um grupo de replicação

Há momentos em que você pode querer visualizar os detalhes de um grupo de replicação. Você pode usar o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache. O processo do console é diferente para Valkey ou Redis OSS (modo cluster desabilitado) e Valkey ou Redis OSS (modo cluster habilitado).

Visualizar os detalhes de um grupo de replicação

- [Visualização de um Valkey ou Redis OSS \(modo cluster desabilitado\) com réplicas](#)
 - [Visualização de detalhes de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) \(console\)](#)
 - [Visualização de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) \(AWS CLI\)](#)
 - [Visualização de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) \(API do ElastiCache\)](#)

- [Visualização de um grupo de replicação: Valkey ou Redis OSS \(modo cluster habilitado\)](#)
 - [Visualização de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)
 - [Visualização de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(AWS CLI\)](#)
 - [Visualização de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(API do ElastiCache\)](#)
- [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#)
- [Visualização dos detalhes de um grupo de replicação \(API do ElastiCache\)](#)

Visualização de um Valkey ou Redis OSS (modo cluster desabilitado) com réplicas

Você pode visualizar os detalhes de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com réplicas (API/CLI: grupo de replicação) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

Visualização de detalhes de um cluster do Valkey ou Redis OSS (modo cluster desabilitado)

- [Visualização de detalhes de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) \(console\)](#)
- [Visualização de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) \(AWS CLI\)](#)
- [Visualização de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) \(API do ElastiCache\)](#)

Visualização de detalhes de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) (console)

Para ver os detalhes de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com réplicas usando o console do ElastiCache, consulte o tópico [Visualização dos detalhes do Valkey ou Redis OSS \(modo cluster desabilitado\) \(console\)](#).

Visualização de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) (AWS CLI)

Para um exemplo de AWS CLI que exibe os detalhes de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado), consulte [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#).

Visualização de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) (API do ElastiCache)

Para um exemplo da API do ElastiCache que exibe os detalhes de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado), consulte [Visualização dos detalhes de um grupo de replicação \(API do ElastiCache\)](#).

Visualização de um grupo de replicação: Valkey ou Redis OSS (modo cluster habilitado)

Visualização de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (console)

Para ver os detalhes de um cluster do Valkey ou Redis OSS (modo cluster habilitado) usando o console do ElastiCache, consulte [Visualização de detalhes de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#).

Visualização de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (AWS CLI)

Para um exemplo da CLI do ElastiCache que exibe os detalhes de um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado), consulte [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#).

Visualização de um cluster do Valkey ou Redis OSS (modo cluster habilitado) (API do ElastiCache)

Para um exemplo da API do ElastiCache que exibe os detalhes de um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado), consulte [Visualização dos detalhes de um grupo de replicação \(API do ElastiCache\)](#).

Visualização dos detalhes de um grupo de replicação (AWS CLI)

É possível visualizar os detalhes de um grupo de replicação usando o comando AWS CLI da `describe-replication-groups`. Use os seguintes parâmetros opcionais para refinar a listagem. A omissão dos parâmetros retorna os detalhes de até 100 grupos de replicação.

Parâmetros opcionais

- `--replication-group-id`: use este parâmetro para listar os detalhes de um grupo de replicação específico. Se o grupo de replicação especificado tiver mais de um grupo de nós, os resultados serão retornados agrupados por grupo de nós.
- `--max-items`: use este parâmetro para limitar o número de grupos de replicação listados. O valor de `--max-items` não pode ser inferior a 20 ou superior a 100.

Example

O código a seguir lista os detalhes para até 100 grupos de replicação.

```
aws elasticache describe-replication-groups
```

O código a seguir lista os detalhes para sample-repl-group.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

O código a seguir lista os detalhes para sample-repl-group.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

O código a seguir lista os detalhes para até 25 grupos de replicação.

```
aws elasticache describe-replication-groups --max-items 25
```

O resultado dessa operação deve ser semelhante a este (formato JSON).

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "test",
      "NodeGroups": [
        {
          "Status": "available",
          "NodeGroupMembers": [
            {
              "CurrentRole": "primary",
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
              },
              "CacheClusterId": "rg-name-001"
            },
            {
              "CurrentRole": "replica",
              "PreferredAvailabilityZone": "us-west-2b",
```

```

        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Port": 6379,
            "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "rg-name-002"
    },
    {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Port": 6379,
            "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "rg-name-003"
    }
],
"NodeGroupId": "0001",
"PrimaryEndpoint": {
    "Port": 6379,
    "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
}
}
],
"ReplicationGroupId": "rg-name",
"AutomaticFailover": "enabled",
"SnapshottingClusterId": "rg-name-002",
"MemberClusters": [
    "rg-name-001",
    "rg-name-002",
    "rg-name-003"
],
"PendingModifiedValues": {}
},
{
    ... some output omitted for brevity
}
]
}

```

Para obter mais informações, consulte a AWS CLI para o tópico do ElastiCache [describe-replication-groups](#).

Visualização dos detalhes de um grupo de replicação (API do ElastiCache)

Você pode visualizar os detalhes de uma replicação usando a operação AWS CLI

`DescribeReplicationGroups`. Use os seguintes parâmetros opcionais para refinar a listagem. A omissão dos parâmetros retorna os detalhes de até 100 grupos de replicação.

Parâmetros opcionais

- `ReplicationGroupId`: use este parâmetro para listar os detalhes de um grupo de replicação específico. Se o grupo de replicação especificado tiver mais de um grupo de nós, os resultados serão retornados agrupados por grupo de nós.
- `MaxRecords`: use este parâmetro para limitar o número de grupos de replicação listados. O valor de `MaxRecords` não pode ser inferior a 20 ou superior a 100. O padrão é 100.

Example

O código a seguir lista os detalhes para até 100 grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

O código a seguir lista os detalhes para `myReplGroup`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=myReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

O código a seguir lista os detalhes para até 25 clusters.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DescribeReplicationGroups
&MaxRecords=25
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte o tópico de referência da API do ElastiCache [DescribeReplicationGroups](#).

Localização de endpoints de grupo de replicação

Um aplicativo pode se conectar a qualquer nó em um grupo de replicação, desde que tenha o endpoint DNS e o número da porta desse nó. Dependendo de você estar executando um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) ou do Valkey ou Redis OSS (modo cluster habilitado), você estará interessado em diferentes endpoints.

Valkey ou Redis OSS (modo cluster desabilitado)

Os clusters Valkey ou Redis OSS (modo cluster desabilitado) com réplicas têm três tipos de endpoints: o endpoint primário, o endpoint leitor e os endpoints de nó. O endpoint primário é um nome DNS que sempre é resolvido para o nó primário no cluster. O endpoint primário é imune às alterações no seu cluster, como promover uma réplica de leitura para a função primária. Para atividades de gravação, recomendamos que suas aplicações se conectem ao endpoint primário.

Um endpoint de leitor divide uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura em um cluster do ElastiCache. Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar várias réplicas de leitura do seu cluster do ElastiCache para Redis OSS em diferentes zonas de disponibilidade da AWS (AZ) para garantir a alta disponibilidade dos endpoints leitores.

Note

Um endpoint leitor não é um balanceador de carga. É um registro DNS que vai resolver para um endereço IP de um dos nós de réplica em uma forma round robin.

Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó no seu aplicativo.

Valkey ou Redis OSS (modo cluster habilitado)

Clusters do Valkey ou Redis OSS (modo cluster habilitado) com réplicas, como eles têm vários fragmentos (API/CLI: grupos de nós), o que significa que também possuem vários nós primários, possuem uma estrutura de endpoint diferente dos clusters do Valkey ou Redis OSS (modo cluster

desabilitado). O Valkey ou Redis OSS (modo cluster habilitado) tem um endpoint de configuração que "conhece" todos os endpoints primários e de nó no cluster. Seu aplicativo conecta-se ao endpoint de configuração. Sempre que seu aplicativo grava ou lê no/do endpoint de configuração do cluster, o Valkey ou Redis OSS, nos bastidores, determina a qual fragmento a chave pertence e qual endpoint nesse fragmento deve ser usado. O processo é bastante transparente para o seu aplicativo.

Você pode localizar os endpoints de um cluster usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Localizar endpoints de grupo de replicação

Para localizar os endpoints do seu grupo de replicação, consulte um dos seguintes tópicos:

- [Localização de endpoints de um cluster do Valkey ou do Redis OSS \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para um cluster do Valkey ou do Redis OSS \(modo cluster habilitado\) \(console\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação do Valkey ou Redis OSS \(API do ElastiCache\)](#)

Modificação de um grupo de replicação

Restrições importantes

- Atualmente, ElastiCache oferece suporte a modificações limitadas de um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado), por exemplo, alterando a versão do mecanismo usando a operação de API (`ModifyReplicationGroupCL`). `modify-replication-group` Você pode modificar o número de fragmentos (grupos de nós) em um cluster do Valkey ou Redis OSS (modo cluster habilitado) com a operação da API [ModifyReplicationGroupShardConfiguration](#) (CLI: [modify-replication-group-shard-configuration](#)). Para obter mais informações, consulte [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#).

Outras modificações em um cluster do Valkey ou Redis OSS (modo cluster habilitado) exigem que você o crie novamente com o novo cluster que incorpore as alterações.

- Você pode atualizar os clusters e grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado) e Valkey ou Redis OSS (modo cluster habilitado) para versões mais recentes do mecanismo. No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente ou grupo de replicação e criando-o novamente. Para obter mais informações, consulte [Gerenciamento de versões para ElastiCache](#).
- Você pode atualizar um cluster OSS existente ElastiCache para Valkey ou Redis que usa o modo de cluster desativado para usar o modo de cluster ativado, usando o console, a [ModifyReplicationGroup](#) API ou o comando `modify-replication-group` CLI, conforme mostrado no exemplo abaixo. Ou você pode seguir as etapas em [Modificar o modo de cluster](#).

Você pode modificar as configurações de um cluster Valkey ou Redis OSS (modo de cluster desativado) usando o ElastiCache console AWS CLI, o ou a API. Atualmente, ElastiCache oferece suporte a um número limitado de modificações em um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). Outras modificações exigem que você crie um backup do grupo de replicação e, em seguida, use esse backup para propagar um novo grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado).

Tópicos

- [Usando o Console de gerenciamento da AWS](#)
- [Usando o AWS CLI](#)
- [Usando a ElastiCache API](#)

Usando o Console de gerenciamento da AWS

Para modificar um cluster do Valkey ou Redis OSS (modo cluster desabilitado), consulte [Modificação de um cluster do ElastiCache](#).

Usando o AWS CLI

Veja a seguir AWS CLI exemplos do `modify-replication-group` comando. Você pode usar o mesmo comando para fazer outras modificações em um grupo de replicação.

Ativar multi-AZ em um grupo de replicação do Valkey ou Redis OSS existente:

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --multi-az-enabled = true
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --multi-az-enabled
```

Modifique o modo de cluster de desativado para ativado:

Para modificar o modo de cluster de desativado para ativado, você deve primeiro definir o modo de cluster como compatível. O modo compatível permite que seus clientes Valkey ou Redis OSS se conectem usando o modo de cluster habilitado e o modo de cluster desabilitado. Depois de migrar todos os clientes Valkey ou Redis OSS para usar o modo cluster habilitado, você poderá concluir a configuração do modo cluster e definir o modo cluster como Habilitado.

Para Linux, macOS ou Unix:

Defina o modo de cluster como compatível.

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --cache-parameter-group-name myParameterGroupName \  
  --cluster-mode compatible
```

Defina o modo de cluster como ativado.

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --cluster-mode enabled
```

Para Windows:

Defina o modo de cluster como compatível.

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --cache-parameter-group-name myParameterGroupName ^  
  --cluster-mode compatible
```

Defina o modo de cluster como ativado.

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --cluster-mode enabled
```

Para obter mais informações sobre o AWS CLI `modify-replication-group` comando, consulte [modify-replication-group](#) ou [Modificando o modo de cluster](#) no Guia do ElastiCache usuário do Redis OSS.

Usando a ElastiCache API

A operação de ElastiCache API a seguir habilita o Multi-AZ em um grupo de replicação Valkey ou Redis OSS existente. Você pode usar a mesma operação para fazer outras modificações em um grupo de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &AutomaticFailoverEnabled=true  
  &Mutli-AZEnabled=true
```

```
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Para obter mais informações sobre a `ModifyReplicationGroup` operação ElastiCache da API, consulte [ModifyReplicationGroup](#).

Exclusão de um grupo de replicação

Se você não precisar mais de um dos seus clusters com réplicas (chamados de grupos de replicação na API/CLI), poderá excluí-lo. Quando você exclui um grupo de replicação, o ElastiCache exclui todos os nós nesse grupo.

Depois de ter iniciado essa operação, ela não poderá ser interrompida ou cancelada.

Warning

- Ao excluir um cluster do ElastiCache for Redis OSS, seus snapshots manuais são retidos. Você também poderá criar um snapshot final antes da exclusão do cluster. Os snapshots de cache automáticos não são retidos.
- É necessário ter a permissão `CreateSnapshot` para criar um snapshot final. Sem essa permissão, a chamada de API falhará com uma exceção `Access Denied`.

Exclusão de um grupo de replicação (Console)

Para excluir um cluster que possui réplicas, consulte [Excluindo um cluster no ElastiCache](#).

Excluir um grupo de replicação (AWS CLI)

Use o comando [delete-replication-group](#) para excluir um grupo de replicação.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Um prompt solicita que você confirme sua decisão. Insira y (sim) para iniciar a operação imediatamente. Após o início do processo, ele é irreversível.

```
After you begin deleting this replication group, all of its nodes will be deleted as well.
```

```
Are you sure you want to delete this replication group? [Ny]y
```

```
REPLICATIONGROUP my-repgroup My replication group deleting
```

Exclusão de um grupo de replicação (API do ElastiCache)

Chame [DeleteReplicationGroup](#) com o parâmetro `ReplicationGroup`.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteReplicationGroup  
&ReplicationGroupId=my-repgroup  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Note

Se você definir o parâmetro `RetainPrimaryCluster` como `true`, todas as réplicas de leitura serão excluídas, mas o cluster primário será mantido.

Alteração do número de réplicas

Você pode aumentar ou diminuir dinamicamente o número de réplicas de leitura no grupo de replicação do Valkey ou Redis OSS usando o Console de gerenciamento da AWS, a AWS CLI ou a API do ElastiCache. Caso seu grupo de replicação seja um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado), você pode escolher de quais fragmentos (grupos de nós) aumentar ou diminuir o número de réplicas.

Para alterar dinamicamente o número de réplicas no grupo de replicação, escolha a operação na tabela a seguir que se adapte à situação.

Para fazer isso	Para Valkey ou Redis OSS (modo cluster habilitado)	Para Valkey ou Redis OSS (modo cluster desabilitado)
Adicionar réplicas	Aumento do número de réplicas em um fragmento	Aumento do número de réplicas em um fragmento Adição de uma réplica de leitura do Valkey ou do Redis OSS (modo cluster desabilitado)
Excluir réplicas	Diminuição do número de réplicas em um fragmento	Diminuição do número de réplicas em um fragmento Exclusão de uma réplica de leitura do Valkey ou do Redis OSS (modo cluster desabilitado)

Aumento do número de réplicas em um fragmento

Você pode aumentar o número de réplicas em um fragmento do Valkey ou Redis OSS (modo cluster habilitado) ou grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) até um máximo de cinco. Você pode fazer isso usando o Console de gerenciamento da AWS, a AWS CLI ou a API do ElastiCache.

Tópicos

- [Utilização do Console de gerenciamento da AWS](#)
- [Como usar o AWS CLI](#)
- [Uso da API do ElastiCache](#)

Utilização do Console de gerenciamento da AWS

O procedimento a seguir usa o console para aumentar o número de réplicas em um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado).

Para aumentar o número de réplicas em fragmentos

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS e o nome do grupo de replicação ao qual você deseja adicionar réplicas.
3. Escolha a caixa de cada estilhaço a que você deseja adicionar réplicas.
4. Escolha Add replicas (Adicionar réplicas).
5. Conclua a página Add Replicas to Shards (Adicionar réplicas a estilhaços):
 - Em New number of replicas/shard (Novo número de réplicas/estilhaço), informe o número de réplicas que todos os estilhaços selecionados devem ter. Esse valor deve ser maior que ou igual a Current Number of Replicas per shard (Número atual de réplicas por estilhaço) e menor que ou igual a cinco. Recomendamos pelo menos duas réplicas para trabalho mínimo.
 - Em Availability Zones (Zonas de disponibilidade), escolha No preference (Sem preferência) para o ElastiCache escolher uma zona de disponibilidade para cada nova réplica, ou Specify Availability Zones (Especificar zonas de disponibilidade) a fim de escolher uma zona de disponibilidade para cada nova réplica.

Caso você escolha Specify Availability Zones (Especificar zonas de disponibilidade), para cada nova réplica, especifique uma zona de disponibilidade usando a lista.

6. Escolha Add (Adicionar) a fim de adicionar as réplicas ou Cancel (Cancelar) a fim de cancelar a operação.

Como usar o AWS CLI

Para aumentar o número de réplicas em um fragmento do Valkey ou Redis OSS, use o comando `increase-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: obrigatório. Identifica em qual grupo de replicação você deseja aumentar o número de réplicas.
- `--apply-immediately` ou `--no-apply-immediately`: obrigatório. Especifica se é necessário aumentar a contagem de réplicas imediatamente (`--apply-immediately`) ou na próxima janela de manutenção (`--no-apply-immediately`). Atualmente, `--no-apply-immediately` não é suportado.
- `--new-replica-count`: opcional. Especifica o número de nós de réplica que você deseja concluir, até um máximo de cinco. Use este parâmetro em grupos de replicação do Valkey ou do Redis OSS (modo cluster desabilitado) nos quais haja apenas um grupo de nós ou o grupo do Valkey ou do Redis OSS (modo cluster habilitado), ou onde você deseja que todos os grupos de nós tenham o mesmo número de réplicas. Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- `--replica-configuration`: opcional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use este parâmetro em grupos do Valkey ou Redis OSS (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

`--replica-configuration` tem três membros opcionais:

- `NodeId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado), o ID de fragmento é sempre `0001`. Para encontrar o ID de um grupo de nós (fragmento) do Valkey ou Redis OSS (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- `NewReplicaCount`: o número de réplicas que você deseja nesse grupo de nós ao final desta operação. O valor deve ser maior que o número atual de réplicas, até um máximo de cinco.

Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.

- `PreferredAvailabilityZones`: uma lista de strings de `PreferredAvailabilityZone` que especificam em quais zonas de disponibilidade os nós do grupo de replicação devem estar. O número de valores `PreferredAvailabilityZone` deve ser igual ao valor de `NewReplicaCount` mais 1 para contabilizar o nó primário. Caso esse membro de `--replica-configuration` seja omitido, o ElastiCache para Redis OSS escolherá a zona de disponibilidade para cada uma das novas réplicas.

Important

Você deve incluir o parâmetro `--new-replica-count` ou `--replica-configuration`, mas não ambos, na chamada.

Example

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para três. Quando o exemplo é concluído, existem três réplicas em cada grupo de nós. Esse número se aplica caso este seja um grupo do Valkey ou Redis OSS (modo cluster desabilitado) com um grupo de nós único ou um grupo do Valkey ou Redis OSS (modo cluster habilitado) com vários grupos de nós.

Para Linux, macOS ou Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id sample-repl-group \  
  --new-replica-count 3 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --new-replica-count 3 ^  
  --apply-immediately
```

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

Para Linux, macOS ou Unix:

```
aws elasticache increase-replica-count \
  --replication-group-id sample-repl-group \
  --replica-configuration \
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
  --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^
  --replication-group-id sample-repl-group ^
  --replica-configuration ^
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
  --apply-immediately
```

Para obter mais informações sobre como aumentar o número de réplicas usando a CLI, consulte [increase-replica-count](#) na Referência da linha de comando do Amazon ElastiCache.

Uso da API do ElastiCache

Para aumentar o número de réplicas em um fragmento do Valkey ou Redis OSS, use a ação `IncreaseReplicaCount` com os seguintes parâmetros:

- `ReplicationGroupId`: obrigatório. Identifica em qual grupo de replicação você deseja aumentar o número de réplicas.

- **ApplyImmediately**: obrigatório. Especifica se é necessário aumentar a contagem de réplicas imediatamente (`ApplyImmediately=True`) ou na próxima janela de manutenção (`ApplyImmediately=False`). Atualmente, `ApplyImmediately=False` não é suportado.
- **NewReplicaCount**: opcional. Especifica o número de nós de réplica que você deseja concluir, até um máximo de cinco. Use este parâmetro em grupos de replicação do Valkey ou do Redis OSS (modo cluster desabilitado) nos quais haja apenas um grupo de nós, ou grupos do Valkey ou do Redis OSS (modo cluster habilitado) onde você deseja que todos os grupos de nós tenham o mesmo número de réplicas. Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- **ReplicaConfiguration**: opcional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use este parâmetro em grupos do Valkey ou Redis OSS (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

ReplicaConfiguration tem três membros opcionais:

- **NodeId**: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado), o ID do grupo de nós (fragmento) é sempre 0001. Para encontrar o ID de um grupo de nós (fragmento) do Valkey ou Redis OSS (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- **NewReplicaCount**: o número de réplicas que você deseja nesse grupo de nós ao final desta operação. O valor deve ser maior que o número atual de réplicas e um máximo de cinco. Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- **PreferredAvailabilityZones**: uma lista de strings de `PreferredAvailabilityZone` que especificam em quais zonas de disponibilidade os nós do grupo de replicação devem estar. O número de valores `PreferredAvailabilityZone` deve ser igual ao valor de `NewReplicaCount` mais 1 para contabilizar o nó primário. Caso esse membro de `ReplicaConfiguration` seja omitido, o ElastiCache para Redis OSS escolherá a zona de disponibilidade para cada uma das novas réplicas.

Important

Você deve incluir o parâmetro `NewReplicaCount` ou `ReplicaConfiguration`, mas não ambos, na chamada.

Example

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para três. Quando o exemplo é concluído, existem três réplicas em cada grupo de nós. Esse número se aplica caso este seja um grupo do Valkey ou Redis OSS (modo cluster desabilitado) com um grupo de nós único, ou um grupo do Valkey ou Redis OSS (modo cluster habilitado) com vários grupos de nós.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=True  
  &NewReplicaCount=3  
  &ReplicationGroupId=sample-repl-group  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=True  
  &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001  
  &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=  
east-1a  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=  
east-1c  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=  
east-1b  
  &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
```

```
&ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1c

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
  &ReplicationGroupId=sample-repl-group
  &Version=2015-02-02
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20150202T192317Z
  &X-Amz-Credential=<credential>
```

Para obter mais informações sobre como aumentar o número de réplicas usando a API, consulte [IncreaseReplicaCount](#) na Referência da API do Amazon ElastiCache.

Diminuição do número de réplicas em um fragmento

Você pode reduzir o número de réplicas em um fragmento do Valkey ou do Redis OSS (modo cluster habilitado) ou em um grupo de replicação do Valkey ou do Redis OSS (modo cluster desabilitado):

- Para o Valkey ou Redis OSS (modo cluster desabilitado), é possível reduzir o número de réplicas para um, caso o multi-AZ esteja habilitado, e para zero, caso ele não esteja habilitado.
- Para o Valkey ou Redis OSS (modo cluster habilitado), você pode diminuir o número de réplicas para zero. No entanto, você não pode fazer failover para uma réplica, caso o nó primário falhe.

Você pode usar o Console de gerenciamento da AWS, a AWS CLI ou a API do ElastiCache para reduzir o número de réplicas em um grupo de nós (fragmento) ou em um grupo de replicação.

Tópicos

- [Utilização do Console de gerenciamento da AWS](#)
- [Como usar o AWS CLI](#)
- [Uso da API do ElastiCache](#)

Utilização do Console de gerenciamento da AWS

O procedimento a seguir usa o console para reduzir o número de réplicas em um grupo de replicação do Valkey ou do Redis OSS (modo cluster habilitado).

Para diminuir o número de réplicas em um fragmento do Valkey ou Redis OSS

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis OSS e o nome do grupo de replicação do qual você deseja excluir réplicas.
3. Escolha a caixa de cada estilhaço de que você deseja excluir um nó de réplica.
4. Escolha Delete replicas (Excluir réplicas).
5. Conclua a página Delete Replicas from Shards (Excluir réplicas de estilhaços):
 - a. Em New number of replicas/shard (Novo número de réplicas/estilhaço), informe o número de réplicas que os estilhaços selecionados devem ter. Esse número deve ser maior que ou igual a 1. Recomendamos pelo menos duas réplicas por estilhaço para trabalho mínimo.

- b. Escolha Delete (Excluir) a fim de excluir as réplicas ou Cancel (Cancelar) a fim de cancelar a operação.

 Important

- Caso você não especifique os nós de réplica a serem excluídos, o ElastiCache para Redis OSS selecionará automaticamente os nós de réplica para exclusão. Ao fazer isso, o ElastiCache para Redis OSS tenta manter a arquitetura multi-AZ do grupo de replicação seguido preservando réplicas com atraso de replicação mínimo em relação ao principal.
- Não é possível excluir o primário ou os nós primários em um grupo de replicação. Se você especificar o nó primário para exclusão, a operação falhará com um evento de erro indicando que o nó primário foi selecionado para exclusão.

Como usar o AWS CLI

Para diminuir o número de réplicas em um fragmento do Valkey ou Redis OSS, use o comando `decrease-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: obrigatório. Identifica em qual grupo de replicação você deseja diminuir o número de réplicas.
- `--apply-immediately` ou `--no-apply-immediately`: obrigatório. Especifica se é necessário diminuir a contagem de réplicas imediatamente (`--apply-immediately`) ou na próxima janela de manutenção (`--no-apply-immediately`). Atualmente, `--no-apply-immediately` não é suportado.
- `--new-replica-count`: opcional. Especifica o número de nós de réplica desejado. O valor de `--new-replica-count` deve ser um valor válido menos o número atual de réplicas nos grupos de nós. Para valores permitidos mínimos, consulte [Diminuição do número de réplicas em um fragmento](#). Caso o valor de `--new-replica-count` não atenda a esse requisito, a chamada falha.
- `--replicas-to-remove`: opcional. Contém uma lista de IDs de nó que especificam os nós de réplica a serem removidos.
- `--replica-configuration`: opcional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use este parâmetro em grupos do Valkey ou Redis OSS (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

--replica-configuration tem três membros opcionais:

- **NodeId:** o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado), o ID de fragmento é sempre 0001. Para encontrar o ID de um grupo de nós (fragmento) do Valkey ou Redis OSS (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- **NewReplicaCount:** um parâmetro opcional que especifica o número de nós de réplica desejado. O valor de NewReplicaCount deve ser um valor válido menos o número atual de réplicas nos grupos de nós. Para valores permitidos mínimos, consulte [Diminuição do número de réplicas em um fragmento](#). Caso o valor de NewReplicaCount não atenda a esse requisito, a chama falha.
- **PreferredAvailabilityZones:** uma lista de strings de PreferredAvailabilityZone que especificam em quais zonas de disponibilidade os nós do grupo de replicação estão. O número de valores PreferredAvailabilityZone deve ser igual ao valor de NewReplicaCount mais 1 para contabilizar o nó primário. Caso esse membro de --replica-configuration seja omitido, o ElastiCache para Redis OSS escolherá a zona de disponibilidade para cada uma das novas réplicas.

 Important

Você deve incluir um, e somente um, parâmetro --new-replica-count, --replicas-to-remove, ou --replica-configuration.

Example

O exemplo a seguir usa --new-replica-count para diminuir o número de réplicas no grupo de replicação sample-repl-group para um. Quando o exemplo é concluído, existe uma réplica em cada grupo de nós. Esse número se aplica caso este seja um grupo do Valkey ou Redis OSS (modo cluster desabilitado) com um grupo de nós único ou um grupo do Valkey ou Redis OSS (modo cluster habilitado) com vários grupos de nós.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count
  --replication-group-id sample-repl-group \
  --new-replica-count 1 \
```



```
--apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --new-replica-count 1 ^  
  --apply-immediately
```

O exemplo a seguir diminui o número de réplicas no grupo de replicação *sample-repl-group* removendo duas réplicas especificadas (0001 e 0003) do grupo de nós.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id sample-repl-group \  
  --replicas-to-remove 0001,0003 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --replicas-to-remove 0001,0003 \  
  --apply-immediately
```

O exemplo a seguir usa `--replica-configuration` a fim de diminuir o número de réplicas no grupo de replicação *sample-repl-group* para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id sample-repl-group \  
  --replica-configuration \  
    --replicas-per-availability-zone 1 ^  
    --preferred-availability-zones us-east-1a,us-east-1b ^
```

```

NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-
east-1c \
NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-
east-1b,us-east-1c \
--apply-immediately

```

Para Windows:

```

aws elasticache decrease-replica-count ^
--replication-group-id sample-repl-group ^
--replica-configuration ^
NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-
east-1c ^
NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-
east-1b,us-east-1c \
--apply-immediately

```

Para obter mais informações sobre como reduzir o número de réplicas usando a CLI, consulte [decrease-replica-count](#) na Referência da linha de comando do Amazon ElastiCache.

Uso da API do ElastiCache

Para diminuir o número de réplicas em um fragmento do Valkey ou Redis OSS, use a ação `DecreaseReplicaCount` com os seguintes parâmetros:

- `ReplicationGroupId`: obrigatório. Identifica em qual grupo de replicação você deseja diminuir o número de réplicas.
- `ApplyImmediately`: obrigatório. Especifica se é necessário diminuir a contagem de réplicas imediatamente (`ApplyImmediately=True`) ou na próxima janela de manutenção (`ApplyImmediately=False`). Atualmente, `ApplyImmediately=False` não é suportado.
- `NewReplicaCount`: opcional. Especifica o número de nós de réplica desejado. O valor de `NewReplicaCount` deve ser um valor válido menos o número atual de réplicas nos grupos de nós. Para valores permitidos mínimos, consulte [Diminuição do número de réplicas em um fragmento](#). Caso o valor de `--new-replica-count` não atenda a esse requisito, a chamada falha.
- `ReplicasToRemove`: opcional. Contém uma lista de IDs de nó que especificam os nós de réplica a serem removidos.
- `ReplicaConfiguration`: opcional. Contém uma lista de grupos de nós que permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use

este parâmetro em grupos do Valkey ou Redis OSS (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

ReplicaConfiguraion tem três membros opcionais:

- **NodeId:** o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Valkey ou do Redis OSS (modo cluster desabilitado), o ID do grupo de nós é sempre 0001. Para encontrar o ID de um grupo de nós (fragmento) do Valkey ou Redis OSS (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- **NewReplicaCount:** o número de réplicas que você deseja nesse grupo de nós ao final desta operação. O valor deve ser menor que o número atual de réplicas até um mínimo de 1, caso o Multi-AZ esteja habilitado ou 0 caso o Multi-AZ com failover automático não esteja habilitado. Caso esse valor não seja menor que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- **PreferredAvailabilityZones:** uma lista de strings de PreferredAvailabilityZone que especificam em quais zonas de disponibilidade os nós do grupo de replicação estão. O número de valores PreferredAvailabilityZone deve ser igual ao valor de NewReplicaCount mais 1 para contabilizar o nó primário. Caso esse membro de ReplicaConfiguration seja omitido, o ElastiCache para Redis OSS escolherá a zona de disponibilidade para cada uma das novas réplicas.

Important

Você deve incluir um, e somente um, parâmetro NewReplicaCount, ReplicasToRemove, ou ReplicaConfiguration.

Example

O exemplo a seguir usa NewReplicaCount para diminuir o número de réplicas no grupo de replicação sample-repl-group para um. Quando o exemplo é concluído, existe uma réplica em cada grupo de nós. Esse número se aplica caso este seja um grupo do Valkey ou Redis OSS (modo cluster desabilitado) com um grupo de nós único ou um grupo do Valkey ou Redis OSS (modo cluster habilitado) com vários grupos de nós.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DecreaseReplicaCount  
&ApplyImmediately=True
```

```
&NewReplicaCount=1
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

O exemplo a seguir diminui o número de réplicas no grupo de replicação `sample-repl-group` removendo duas réplicas especificadas (`0001` e `0003`) do grupo de nós.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&ReplicasToRemove.ReplicaToRemove.1=0001
&ReplicasToRemove.ReplicaToRemove.2=0003
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

O exemplo a seguir usa `ReplicaConfiguration` a fim de diminuir o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
&ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a
```

```
&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
    &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
    &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2

    &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

    &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

    &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
    &ReplicationGroupId=sample-repl-group
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Para obter mais informações sobre como reduzir o número de réplicas usando a API, consulte [DecreaseReplicaCount](#) na Referência da API do Amazon ElastiCache.

Adição de uma réplica de leitura do Valkey ou do Redis OSS (modo cluster desabilitado)

As informações no tópico a seguir só se aplicam a grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado).

À medida que seu tráfego de leitura aumentar, talvez você queira espalhar essas leituras em mais nós e reduzir a pressão de leitura em qualquer nó. Neste tópico, você pode descobrir como adicionar uma réplica de leitura a um cluster do Valkey ou Redis OSS (modo cluster desabilitado).

Um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) pode ter no máximo cinco réplicas de leitura. Se você tentar adicionar uma réplica de leitura a um grupo de replicação que já tenha cinco réplicas de leitura, a operação falhará.

Para obter informações sobre como adicionar réplicas a um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado), consulte o seguinte:

- [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#)
- [Aumento do número de réplicas em um fragmento](#)

Você pode adicionar uma réplica de leitura a um cluster do Valkey ou Redis OSS (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Tópicos relacionados

- [Adicionar nós a um ElastiCache cluster](#)
- [Adição de uma réplica de leitura a um grupo de replicação \(AWS CLI\)](#)
- [Adição de uma réplica de leitura a um grupo de replicação usando a API](#)

Adição de uma réplica de leitura a um grupo de replicação (AWS CLI)

Para adicionar uma réplica de leitura a um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado), use o comando `create-cache-cluster` da AWS CLI com o parâmetro `--replication-group-id` para especificar a qual grupo de replicação adicionar o cluster (nó).

O exemplo a seguir cria o cluster `my-read-replica` e o adiciona ao grupo de replicação `my-replication-group`. Os tipos de nós, os grupos de parâmetros, os grupos de segurança, as janelas de manutenção e outras configurações da réplica de leitura são os mesmos dos outros nós em `my-replication-group`.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-read-replica \  
  --replication-group-id my-replication-group
```

Para Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-read-replica ^  
  --replication-group-id my-replication-group
```

Para obter mais informações sobre como adicionar uma réplica de leitura usando a CLI, consulte [create-cache-cluster](#) na Referência da linha de comando do Amazon ElastiCache.

Adição de uma réplica de leitura a um grupo de replicação usando a API

Para adicionar uma réplica de leitura a um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado), use a operação `CreateCacheCluster` do ElastiCache com o parâmetro `ReplicationGroupId` para especificar a qual grupo de replicação adicionar o cluster (nó).

O exemplo a seguir cria o cluster `myReadReplica` e o adiciona ao grupo de replicação `myReplicationGroup`. Os tipos de nós, os grupos de parâmetros, os grupos de segurança, as janelas de manutenção e outras configurações da réplica de leitura são os mesmos dos outros nós `myReplicationGroup`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=myReadReplica  
&ReplicationGroupId=myReplicationGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações sobre como adicionar uma réplica de leitura usando a API, consulte [CreateCacheCluster](#) na Referência da API do Amazon ElastiCache.

Exclusão de uma réplica de leitura do Valkey ou do Redis OSS (modo cluster desabilitado)

As informações no tópico a seguir só se aplicam a grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado).

À medida que o tráfego de leitura no seu grupo de replicação Valkey ou Redis OSS mudar, talvez você queira adicionar ou remover réplicas de leitura. Remover um nó de um grupo de replicação é o mesmo que simplesmente excluir um cluster, embora existam restrições:

- Não é possível remover o primário de um grupo de replicação. Se quiser excluir o primário, você deverá fazer o seguinte:
 1. Promover uma réplica de leitura para primário. Para obter mais informações sobre como promover uma réplica de leitura para primário, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\)](#).
 2. Exclua o primário antigo. Para uma restrição nesse método, consulte o próximo ponto.
- Se o recurso Multi-AZ estiver habilitado em um grupo de replicação, você não poderá remover a última réplica de leitura do grupo de replicação. Nesse caso, faça o seguinte:
 1. Modifique o grupo de replicação desabilitando o Multi-AZ. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
 2. Exclua a réplica de leitura.

Você pode remover uma réplica de leitura de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

Para obter orientações sobre como excluir um cluster de um grupo de replicação Valkey ou Redis OSS, consulte o seguinte:

- [Usando o Console de gerenciamento da AWS](#)
- [Usando o AWS CLI para excluir um ElastiCache cluster](#)
- [Usando a ElastiCache API](#)
- [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#)
- [Diminuição do número de réplicas em um fragmento](#)

Promoção de uma réplica de leitura a primário, para grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado)

As informações no tópico a seguir só se aplicam a grupos de replicação do Valkey ou Redis OSS (modo cluster desabilitado).

Você pode promover uma réplica de leitura de um cluster do Valkey ou Redis OSS (modo cluster desabilitado) a primário usando o Console de gerenciamento da AWS, a AWS CLI ou a API do ElastiCache. Você não pode promover uma réplica de leitura para primária enquanto Multi-AZ com failover automático estiver habilitado no grupo de replicação. Para promover uma réplica do Valkey ou Redis OSS (modo cluster desabilitado) para primário em um grupo de replicação habilitado para multi-AZ, faça o seguinte:

1. Modifique o grupo de replicação para desabilitar Multi-AZ (isso não requer que todos os clusters estejam na mesma zona de disponibilidade). Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
2. Promova a réplica de leitura para primário.
3. Modifique o grupo de replicação para reabilitar o Multi-AZ.

O Multi-AZ não está disponível em grupos de replicação que executam o Valkey ou Redis OSS 2.6.13 ou anterior.

Utilização do Console de gerenciamento da AWS

O procedimento a seguir usa o console para promover um nó de réplica para primária.

Para promover uma réplica de leitura para primário (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Se a réplica que você quiser promover for membro de um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) em que o multi-AZ estiver habilitado, modifique o grupo de replicação para desabilitar o multi-AZ para continuar. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
3. Escolha Valkey ou Redis OSS, na lista de clusters, escolha o grupo de replicação que você deseja modificar. Esse grupo de replicação deve estar executando o mecanismo "Redis", e não o mecanismo "Clustered Redis", e deve ter dois ou mais nós.

4. Na lista de nós, escolha o nó de réplica que você deseja promover para primária, em Actions (Ações), escolha Promote (Promover).
5. Na caixa de diálogo Promote Read Replica (Promover réplica de leitura), faça o seguinte:
 - a. Em Apply Immediately (Aplicar imediatamente), escolha Yes (Sim) para promover a réplica de leitura imediatamente, ou No (Não) para promovê-la na próxima janela de manutenção do cluster.
 - b. Escolha Promote para promover a réplica de leitura ou Cancel para cancelar a operação.
6. Caso o cluster tenha Multi-AZ habilitado antes de você começar o processo de promoção, aguarde o status do grupo de replicação estar available (disponível), modifique o cluster para reabilitar Multi-AZ. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

Usar a AWS CLI

Você não pode promover uma réplica de leitura para primária caso o grupo de replicação esteja habilitado para Multi-AZ. Em alguns casos, a réplica que você deseja promover pode ser um membro de um grupo de replicação onde o recurso do Multi-AZ está habilitado. Nesses casos, você deve modificar o grupo de replicação para desabilitar o recurso do Multi-AZ antes de continuar. Isso não requer que todos os clusters estejam na mesma zona de disponibilidade. Para obter mais informações sobre como modificar um grupo de replicação, consulte [Modificação de um grupo de replicação](#).

O seguinte comando AWS CLI modifica o grupo de replicação `sample-repl-group`, tornando a réplica de leitura `my-replica-1` o primário no grupo de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id sample-repl-group \  
  --primary-cluster-id my-replica-1
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --primary-cluster-id my-replica-1
```

Para obter mais informações sobre como modificar um grupo de replicação, consulte [modify-replication-group](#) na Referência da linha de comando do Amazon ElastiCache.

Uso da API do ElastiCache

Você não pode promover uma réplica de leitura para primária caso o grupo de replicação esteja habilitado para Multi-AZ. Em alguns casos, a réplica que você deseja promover pode ser um membro de um grupo de replicação onde o recurso do Multi-AZ está habilitado. Nesses casos, você deve modificar o grupo de replicação para desabilitar o recurso do Multi-AZ antes de continuar. Isso não requer que todos os clusters estejam na mesma zona de disponibilidade. Para obter mais informações sobre como modificar um grupo de replicação, consulte [Modificação de um grupo de replicação](#).

A seguinte ação da API do ElastiCache modifica o grupo de replicação myReplGroup, tornando a réplica de leitura myReplica-1 o primário no grupo de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ReplicationGroupId=myReplGroup  
&PrimaryClusterId=myReplica-1  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações sobre como modificar um grupo de replicação, consulte [ModifyReplicationGroup](#) na Referência da API do Amazon ElastiCache.

Gerenciamento de manutenção de cluster do ElastiCache

Cada cluster tem uma janela de manutenção semanal durante a qual todas as alterações do sistema são aplicadas. Com o Valkey e o Redis OSS, os grupos de replicação têm essa mesma janela de manutenção semanal. Se você não especificar uma janela de manutenção preferencial ao criar ou modificar um cluster ou grupo de replicação, o ElastiCache atribuirá uma janela de manutenção

de 60 minutos dentro da janela de manutenção de sua região em um dia da semana escolhido aleatoriamente.

A janela de manutenção de 60 minutos é escolhida aleatoriamente entre um período de 8 horas por região. A tabela a seguir lista os blocos de tempo de cada região dos quais as janelas de manutenção padrão são atribuídas. Você pode escolher uma janela de manutenção preferida fora do bloco de janelas de manutenção da região.

Código da região	Nome da Região	Janela de manutenção da região
ap-northeast-1	Região Ásia-Pacífico (Tóquio)	13h às 21h (UTC)
ap-northeast-2	Região Ásia-Pacífico (Seul)	12h às 20h (UTC)
ap-northeast-3	Região Ásia-Pacífico (Osaka)	12h às 20h (UTC)
ap-southeast-3	Região Ásia-Pacífico (Jacarta)	Das 14:00 às 22:00 (UTC)
ap-south-1	Região Ásia-Pacífico (Mumbai)	17h30 à 1h30 UTC
ap-southeast-1	Região Ásia-Pacífico (Singapura)	Das 14:00 às 22:00 (UTC)
cn-north-1	Região China (Pequim)	14h às 22h (UTC)
cn-northwest-1	Região China (Ningxia)	Das 14:00 às 22:00 (UTC)
ap-east-1	Região Ásia-Pacífico (Hong Kong)	Das 13h às 21h (UTC)
ap-southeast-2	Ásia-Pacífico (Sydney)	12h às 20h (UTC)
eu-west-3	Região Europa (Paris)	De 23:59 a 07:29 UTC
af-south-1	Região África (Cidade do Cabo)	Das 13h às 21h (UTC)
eu-central-1	Região Europa (Frankfurt)	23h às 7h (UTC)
eu-west-1	Região Europa (Irlanda)	22h às 6h (UTC)
eu-west-2	Região Europa (Londres)	23h às 7h (UTC)
me-south-1	Região Oriente Médio (Bahrein)	Das 13h às 21h (UTC)

Código da região	Nome da Região	Janela de manutenção da região
me-central-1	Região do Oriente Médio (Emirados Árabes Unidos)	Das 13h às 21h (UTC)
eu-south-1	Região Europa (Milão)	Das 21:00 às 05:00 (UTC)
sa-east-1	Região América do Sul (São Paulo)	1h às 9h UTC
us-east-1	Região Leste dos EUA (Norte da Virgínia)	3h às 7h (UTC)
us-east-2	Região Leste dos EUA (Ohio)	5h às 12h UTC
us-gov-west-1	AWS GovCloud (US) region	6h às 14h (UTC)
us-west-1	Região Oeste dos EUA (Norte da Califórnia)	6h às 14h (UTC)
us-west-2	Região Oeste dos EUA (Oregon)	6h às 14h (UTC)

Como alterar a janela de manutenção do cluster ou grupo de replicação

A janela de manutenção deve ser definida no horário de menor utilização e, portanto, talvez precise ser modificada de vez em quando. Você pode modificar o cluster ou grupo de replicação para especificar um intervalo de tempo de até 24 horas de duração durante o qual todas as atividades de manutenção solicitadas devem ocorrer. Todas as modificações de cluster diferidas ou pendentes que você tiver solicitado ocorrem durante esse período.

Note

Se você quiser aplicar modificações de tipo de nó e/ou atualizações de mecanismo imediatamente usando o Console de gerenciamento da AWS, selecione a caixa Aplicar agora. Do contrário, essas modificações serão aplicadas durante o próximo período de manutenção programada. Para usar a API, consulte [modify-replication-group](#) ou [modify-cache-cluster](#).

Mais informações

Para obter informações sobre sua janela de manutenção e substituição de nó, consulte:

- [Manutenção do ElastiCache](#): perguntas frequentes sobre manutenção e substituição de nó
- [Substituição de nós \(Memcached\)](#): gerenciamento de substituição de nó para o Memcached
- [Modificação de um cluster do ElastiCache](#): Alteração da janela de manutenção de um cluster
- [Substituição de nós \(Valkey e Redis OSS\)](#): Gerenciamento de substituição de nó
- [Modificação de um grupo de replicação](#): Alteração da janela de manutenção de um grupo de replicação

Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache

O Amazon ElastiCache usa parâmetros para controlar as propriedades de tempo de execução de seus nós e clusters. Geralmente, as versões mais recentes do mecanismo incluem parâmetros adicionais para dar suporte à funcionalidade mais recente. Para tabelas de parâmetros do Memcached, consulte [Parâmetros específicos do Memcached](#). Para tabelas de parâmetros do Valkey e do Redis OSS, consulte [Parâmetros do Valkey e do Redis OSS](#).

Como seria de se esperar, alguns valores de parâmetros, como `maxmemory`, são determinados pelo mecanismo e tipo de nó. Para acessar uma tabela desses valores de parâmetro do Memcached por tipo de nó, consulte [Parâmetros específicos do tipo de nó do Memcached](#). Para acessar uma tabela desses valores de parâmetros do Valkey e do Redis OSS por tipo de nó, consulte [Parâmetros específicos de node-type do Redis OSS](#).

Note

Para obter uma lista de parâmetros específicos do Memcached, consulte [Parâmetros específicos de Memcached](#).

Tópicos

- [Gerenciamento de parâmetros no ElastiCache](#)
- [Camadas do grupo de parâmetros de cache no ElastiCache](#)
- [Criação de um grupo de parâmetros do ElastiCache](#)

- [Listagem de grupos de parâmetros do ElastiCache por nome](#)
- [Listagem dos valores de um grupo de parâmetros do ElastiCache](#)
- [Modificação de um grupo de parâmetros do ElastiCache](#)
- [Exclusão de um grupo de parâmetros do ElastiCache](#)
- [Parâmetros específicos do mecanismo](#)

Gerenciamento de parâmetros no ElastiCache

Os parâmetros do ElastiCache são agrupados em grupos de parâmetros nomeados para facilitar o gerenciamento de parâmetros. Um parameter group representa uma combinação de valores específicos para os parâmetros que são transmitidos ao software do mecanismo durante a inicialização. Esses valores determinam como o processo do mecanismo em cada nó se comportará em runtime. Os valores dos parâmetros em um parameter group específico aplicam-se a todos os nós associados ao grupo, independentemente do cluster ao qual eles pertencem.

Para ajustar o desempenho do cluster, você pode modificar alguns valores de parâmetros ou alterar o parameter group do cluster.

- Não é possível modificar ou excluir os parameter groups padrão. Se você precisar de valores de parâmetros personalizados, deverá criar um parameter group personalizado.
- Para o Memcached, a família do grupo de parâmetros e o cluster ao qual você a está atribuindo devem ser compatíveis. Por exemplo, se seu cluster estiver executando o Memcached versão 1.4.8, você só poderá usar parameter groups, padrão ou personalizados, da família Memcached 1.4.

Para o Redis OSS, a família do grupo de parâmetros e o cluster ao qual você a está atribuindo devem ser compatíveis. Por exemplo, se seu cluster estiver executando o Redis OSS versão 3.2.10, você só poderá usar grupos de parâmetros, padrão ou personalizados, da família Redis OSS 3.2.

- Se você alterar o parameter group de um cluster, os valores de qualquer parâmetro modificável condicionalmente deverão ser os mesmos nos parameter groups atuais e novos.
- Para o Memcached, quando você altera os parâmetros de um cluster, a alteração é aplicada ao cluster imediatamente. Isso é verdadeiro se você alterar o próprio grupo de parâmetro do cluster ou um valor do parâmetro dentro do grupo do parâmetro do cluster. Para determinar quando uma alteração de parâmetro específica é aplicada, consulte a coluna *As alterações terão efeito* nas tabelas para [Parâmetros específicos do Memcached](#). Para obter informações sobre como reinicializar nós de um cluster, consulte [Rebooting clusters](#).
- Para o Redis OSS, quando você altera parâmetros de um cluster, a alteração é aplicada no cluster imediatamente ou, com as exceções indicadas adiante, após os nós do cluster serem reinicializados. Isso é verdadeiro se você alterar o próprio grupo de parâmetro do cluster ou um valor do parâmetro dentro do grupo do parâmetro do cluster. Para determinar quando uma alteração de parâmetro específica é aplicada, consulte a coluna *As alterações terão efeito* nas tabelas para [Parâmetros do Valkey e do Redis OSS](#).

Para obter mais informações sobre a reinicialização de nós do Valkey ou Redis OSS, consulte [Reinicialização de nós](#).

i Alterações de parâmetro do Valkey ou Redis OSS (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster do Valkey ou Redis OSS (modo cluster habilitado), siga as etapas a seguir.

- `activeresharding`
 - bancos de dados
1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
 2. Excluir o cluster. Consulte [Deleting clusters](#).
 3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

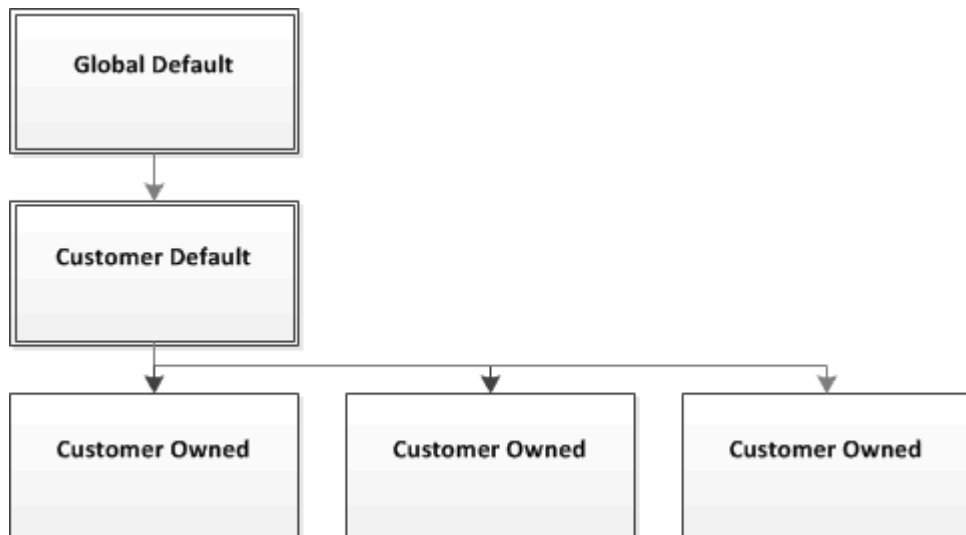
- Você pode associar grupos de parâmetros a datastores globais do Valkey e do Redis OSS. Os datastores globais são uma coleção de um ou mais clusters que abrangem regiões da AWS. Nesse caso, o grupo de parâmetros é compartilhado por todos os clusters que compõem o datastore global. Quaisquer modificações no grupo de parâmetros do cluster primário são replicadas para todos os clusters restantes no datastore global. Para obter mais informações, consulte [Replicação entre AWS regiões usando datastores globais](#).

Você pode verificar se um grupo de parâmetros faz parte de um datastore global procurando nestes locais:

- No console do ElastiCache, na página Parameter Groups (Grupos de parâmetros), o atributo Global yes/no (sim/não).
- A propriedade `IsGlobal` yes/no (sim/não) da operação da API [CacheParameterGroup](#)

Camadas do grupo de parâmetros de cache no ElastiCache

O Amazon ElastiCache tem três níveis de grupos de parâmetros de cache, como mostrado a seguir.



Camadas do grupo de parâmetros do Amazon ElastiCache

Padrão global

O grupo de parâmetros raiz de nível superior para todos os clientes do Amazon ElastiCache na região.

O parameter group de cache padrão global:

- É reservado para o ElastiCache e não está disponível para o cliente.

Padrão do cliente

Uma cópia do parameter group de cache padrão global que é criado para uso do cliente.

O parameter group de cache padrão do cliente:

- É criado e de propriedade do ElastiCache.
- Está disponível ao cliente para ser usado como um parameter group de cache para quaisquer clusters executando uma versão do mecanismo com suporte por esse parameter group de cache.
- Não pode ser editado pelo cliente.

Propriedade do cliente

Uma cópia do parameter group de cache padrão do cliente. Um parameter group de cache do cliente é criado sempre que o cliente cria um parameter group de cache.

O parameter group de cache do cliente mantido:

- É criado e de propriedade do cliente.
- Pode ser atribuído a qualquer um dos clusters compatíveis com o cliente.
- Pode ser modificado pelo cliente para criar um parameter group de cache personalizado.

Nem todos os valores dos parâmetros podem ser modificados. Para obter mais informações sobre valores do Memcached, consulte [Parâmetros específicos do Memcached](#). Para obter mais informações sobre valores do Valkey e do Redis OSS, consulte [Parâmetros do Valkey e do Redis OSS](#).

Criação de um grupo de parâmetros do ElastiCache

Você precisará criar um novo parameter group se houver um ou mais valores de parâmetros que você deseja alterar a partir dos valores padrão. Você pode criar um grupo de parâmetros usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Criação de um grupo de parâmetros do ElastiCache (console)

O procedimento a seguir mostra como criar um grupo de parâmetros usando o console do ElastiCache.

Para criar um grupo de parâmetros usando o console do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Para criar um parameter group, escolha Create Parameter Group.

A tela Create Parameter Group (Criar grupo de parâmetros) será exibida.

4. Na lista Family, escolha a família de parameter groups que será o modelo para o seu parameter group.

A família de grupo de parâmetros, como memcached1.4 ou redis3.2, define os parâmetros efetivos em seu grupo de parâmetros, além de seus valores iniciais. A família de parameter groups deve corresponder ao mecanismo e à versão do cluster.

5. Na caixa Name, digite um nome exclusivo para esse parameter group.

Ao criar um cluster ou modificar o parameter group de um cluster, você escolherá o parameter group pelo seu nome. Portanto, recomendamos que o nome seja informativo e de alguma forma identifique a família do parameter group.

As limitações de nomenclatura de grupo de parâmetros são as seguintes:

- Deve começar com uma letra ASCII.
 - Pode conter apenas letras ASCII, dígitos e hífens.
 - Deve ter entre 1 e 255 caracteres.
 - Não podem conter dois hífens consecutivos.
 - Não podem terminar com um hífen.
6. Na caixa Description, digite uma descrição para o parameter group.
 7. Para criar o parameter group, escolha Create.

Para encerrar o processo sem criar o parameter group, escolha Cancel.

8. Quando o parameter group for criado, ele terá os valores padrão da família. Para alterar os valores padrão, você deve modificar o parameter group. Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

Criação de um grupo de parâmetros do ElastiCache (AWS CLI)

Para criar um grupo de parâmetros usando a AWS CLI, use o comando `create-cache-parameter-group` com esses parâmetros.

- `--cache-parameter-group-name`: O nome do grupo de parâmetros.

As limitações de nomenclatura de grupo de parâmetros são as seguintes:

- Deve começar com uma letra ASCII.
- Pode conter apenas letras ASCII, dígitos e hífens.
- Deve ter entre 1 e 255 caracteres.
- Não podem conter dois hífens consecutivos.

- Não podem terminar com um hífen.
- `--cache-parameter-group-family`: o mecanismo e a família de versões para o grupo de parâmetros.
- `--description`: uma descrição fornecida pelo usuário para o grupo de parâmetros.

Example

O exemplo a seguir cria um parameter group chamado myMem14 usando a família memcached1.4 como modelo.

Para 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"
```

Para Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --cache-parameter-group-family memcached1.4 ^  
  --description "My first parameter group"
```

A saída desse comando deve ser semelhante a esta.

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myMem14",  
    "CacheParameterGroupFamily": "memcached1.4",  
    "Description": "My first parameter group"  
  }  
}
```

Example

O exemplo a seguir cria um parameter group chamado myRed28 usando a família redis2.8 como modelo.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myRed28 \  
  --cache-parameter-group-family redis2.8 \  
  --description "My first parameter group"
```

Para Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myRed28 ^  
  --cache-parameter-group-family redis2.8 ^  
  --description "My first parameter group"
```

A saída desse comando deve ser semelhante a esta.

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myRed28",  
    "CacheParameterGroupFamily": "redis2.8",  
    "Description": "My first parameter group"  
  }  
}
```

Quando o parameter group for criado, ele terá os valores padrão da família. Para alterar os valores padrão, você deve modificar o parameter group. Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

Para obter mais informações, consulte [create-cache-parameter-group](#).

Criação de um grupo de parâmetros do ElastiCache (API do ElastiCache)

Para criar um grupo de parâmetros usando a API do ElastiCache, use a ação `CreateCacheParameterGroup` com esses parâmetros.

- `ParameterGroupName`: O nome do grupo de parâmetros.

As limitações de nomenclatura de grupo de parâmetros são as seguintes:

- Deve começar com uma letra ASCII.
- Pode conter apenas letras ASCII, dígitos e hífen.
- Deve ter entre 1 e 255 caracteres.

- Não podem conter dois hífens consecutivos.
- Não podem terminar com um hífen.
- CacheParameterGroupFamily: o mecanismo e a família de versões para o grupo de parâmetros. Por exemplo, memcached1.4.
- CacheParameterGroupFamily: o mecanismo e a família de versões para o grupo de parâmetros. Por exemplo, redis2.8.
- Description: uma descrição fornecida pelo usuário para o grupo de parâmetros.

Example

O exemplo a seguir cria um parameter group chamado myMem14 usando a família memcached1.4 como modelo.

```
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>
```

A resposta dessa ação deve ser algo semelhante ao seguinte.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <CreateCacheParameterGroupResult>  
    <CacheParameterGroup>  
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>  
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>  
      <Description>My first parameter group</Description>  
    </CacheParameterGroup>  
  </CreateCacheParameterGroupResult>  
  <ResponseMetadata>  
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>  
  </ResponseMetadata>  
</CreateCacheParameterGroupResponse>
```

Example

O exemplo a seguir cria um parameter group chamado myRed28 usando a família redis2.8 como modelo.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheParameterGroup  
&CacheParameterGroupFamily=redis2.8  
&CacheParameterGroupName=myRed28  
&Description=My%20first%20parameter%20group  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

A resposta dessa ação deve ser algo semelhante ao seguinte.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <CreateCacheParameterGroupResult>  
    <CacheParameterGroup>  
      <CacheParameterGroupName>myRed28</CacheParameterGroupName>  
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>  
      <Description>My first parameter group</Description>  
    </CacheParameterGroup>  
  </CreateCacheParameterGroupResult>  
  <ResponseMetadata>  
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>  
  </ResponseMetadata>  
</CreateCacheParameterGroupResponse>
```

Quando o parameter group for criado, ele terá os valores padrão da família. Para alterar os valores padrão, você deve modificar o parameter group. Para obter mais informações, consulte [Modificação de um grupo de parâmetros do ElastiCache](#).

Para obter mais informações, consulte [CreateCacheParameterGroup](#).

Listagem de grupos de parâmetros do ElastiCache por nome

Você pode listar os grupos de parâmetros usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Listagem de grupos de parâmetros por nome (console)

O procedimento a seguir mostra como visualizar uma lista dos grupos de parâmetros usando o console do ElastiCache.

Para listar grupos de parâmetros usando o console do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.

Listagem de grupos de parâmetros do ElastiCache por nome (AWS CLI)

Para gerar uma lista de grupos de parâmetros usando a AWS CLI, use o comando `describe-cache-parameter-groups`. Se você fornecer um nome de parameter group, somente esse parameter group será listado. Se você não fornecer o nome de um parameter group, até `--max-records` parameter groups serão listados. Em ambos os casos, o nome, a família e a descrição do parameter group estão listados.

Example

O código de exemplo a seguir lista o parameter group `myMem14`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name myMem14
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^
  --cache-parameter-group-name myMem14
```

A saída desse comando será algo assim, listando o nome, a família e a descrição do parameter group.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myMem14",
      "CacheParameterGroupFamily": "memcached1.4",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

O código de exemplo a seguir lista o parameter group myRed28.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name myRed28
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^
  --cache-parameter-group-name myRed28
```

A saída desse comando será algo assim, listando o nome, a família e a descrição do parameter group.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed28",
      "CacheParameterGroupFamily": "redis2.8",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

O código de exemplo a seguir lista o grupo de parâmetros myRed56 para grupos de parâmetros em execução no mecanismo Redis OSS versão 5.0.6 e posteriores. Se o grupo de parâmetros fizer parte

de um [Replicação entre AWS regiões usando datastores globais](#), o valor da propriedade `IsGlobal` retornado na saída será `Yes`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myRed56
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myRed56
```

A saída desse comando será algo semelhante ao seguinte: listagem de nome, família, `isGlobal` e descrição do grupo de parâmetros.

```
{  
  "CacheParameterGroups": [  
    {  
      "CacheParameterGroupName": "myRed56",  
      "CacheParameterGroupFamily": "redis5.0",  
      "Description": "My first parameter group",  
      "IsGlobal": "yes"  
    }  
  ]  
}
```

Example

O código de exemplo a seguir lista até 10 parameter groups.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

A saída JSON desse comando será algo semelhante ao seguinte, listagem de nome, família, descrição e, no caso do `redis5.6`, se o grupo de parâmetros faz parte de um datastore global (`isGlobal`), para cada grupo de parâmetros.

```
{  
  "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"
},
]
}
```

Para obter mais informações, consulte [describe-cache-parameter-groups](#).

Listagem de grupos de parâmetros do ElastiCache por nome (API do ElastiCache)

Para gerar uma lista de grupos de parâmetros usando a API do ElastiCache, use a ação `DescribeCacheParameterGroups`. Se você fornecer um nome de parameter group, somente esse parameter group será listado. Se você não fornecer o nome de um parameter group, até `MaxRecords` parameter groups serão listados. Em ambos os casos, o nome, a família e a descrição do parameter group estão listados.

Example

O código de exemplo a seguir lista o parameter group `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>
```

A resposta dessa ação parecerá algo semelhante ao seguinte, listando o nome, a família e a descrição de cada parameter group.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParameterGroupsResult>  
    <CacheParameterGroups>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>  
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>  
        <Description>My custom Memcached 1.4 parameter group</Description>  
      </CacheParameterGroup>  
    </CacheParameterGroups>  
  </DescribeCacheParameterGroupsResult>  
  <ResponseMetadata>  
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>  
  </ResponseMetadata>  
</DescribeCacheParameterGroupsResponse>
```

Example

O código de exemplo a seguir lista até 10 parameter groups.

```
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>
```

A resposta dessa ação será semelhante ao seguinte, listagem do nome, família, descrição e, no caso do redis5.6, se o grupo de parâmetros pertence a um datastore (isGlobal), para cada grupo de parâmetros.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParameterGroupsResult>  
    <CacheParameterGroups>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myRedis28</CacheParameterGroupName>  
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>  
        <Description>My custom Redis 2.8 parameter group</Description>  
      </CacheParameterGroup>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>  
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>  
        <Description>My custom Memcached 1.4 parameter group</Description>  
      </CacheParameterGroup>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myRedis56</CacheParameterGroupName>  
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>  
        <Description>My custom redis 5.6 parameter group</Description>  
        <isGlobal>yes</isGlobal>  
      </CacheParameterGroup>  
    </CacheParameterGroups>  
  </DescribeCacheParameterGroupsResult>  
  <ResponseMetadata>  
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>  
  </ResponseMetadata>
```

```
</DescribeCacheParameterGroupsResponse>
```

Example

O código de exemplo a seguir lista o parameter group myRed28.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameterGroups  
&CacheParameterGroupName=myRed28  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

A resposta dessa ação será semelhante ao seguinte: listagem do nome, família e descrição.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParameterGroupsResult>  
    <CacheParameterGroups>  
      <CacheParameterGroup>  
        <CacheParameterGroupName>myRed28</CacheParameterGroupName>  
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>  
        <Description>My custom Redis 2.8 parameter group</Description>  
      </CacheParameterGroup>  
    </CacheParameterGroups>  
  </DescribeCacheParameterGroupsResult>  
  <ResponseMetadata>  
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>  
  </ResponseMetadata>  
</DescribeCacheParameterGroupsResponse>
```

Example

O código de exemplo a seguir lista o grupo de parâmetros myRed56.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameterGroups  
&CacheParameterGroupName=myRed56  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

A resposta desta ação será semelhante ao seguinte, listagem do nome, família, descrição e se o grupo de parâmetros faz parte de um datastore global (isGlobal).

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom Redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Para obter mais informações, consulte [DescribeCacheParameterGroups](#).

Listagem dos valores de um grupo de parâmetros do ElastiCache

Você pode listar os parâmetros e seus valores para um grupo de parâmetros usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Listagem dos valores de um grupo de parâmetros do ElastiCache (console)

O procedimento a seguir mostra como listar os parâmetros e seus valores para um grupo de parâmetros usando o console do ElastiCache.

Para listar os parâmetros de um grupo e seus valores usando o console do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Escolha o parameter group para o qual você deseja listar os parâmetros e os valores, escolhendo a caixa à esquerda do nome do parameter group.

Os parâmetros e seus valores serão listados na parte inferior da tela. Devido ao número de parâmetros, talvez seja necessário rolar para cima e para baixo para encontrar o parâmetro de interesse.

Listagem dos valores de um grupo de parâmetros (AWS CLI)

Para listar os parâmetros de um grupo de parâmetros e seus valores usando a AWS CLI, use o comando `describe-cache-parameters`.

Example

O código de exemplo a seguir lista todos os parâmetros do Memcached e seus valores para o grupo de parâmetros `myMem14`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myMem14
```

Para Windows:

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myMem14
```

Example

O código de exemplo a seguir lista todos os parâmetros e seus valores para o parameter group myRedis28.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myRedis28
```

Para Windows:

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myRed28
```

Para obter mais informações, consulte [describe-cache-parameters](#).

Listagem dos valores de um grupo de parâmetros (API do ElastiCache)

Para listar os parâmetros de um grupo de parâmetros e seus valores usando a API do ElastiCache, use a ação DescribeCacheParameters.

Example

O código de exemplo a seguir lista todos os parâmetros do Memcached para o grupo de parâmetros 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>
```

A resposta dessa ação será semelhante a esta. Essa resposta foi truncada.

```

<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>
        </CacheNodeTypeSpecificValues>
      </CacheNodeTypeSpecificParameter>
    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>
...output omitted...

```

Example

O código de exemplo a seguir lista todos os parâmetros para o parameter group myRed28.

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameters

```

```

&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

A resposta dessa ação será semelhante a esta. Essa resposta foi truncada.

```

<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>
        </CacheNodeTypeSpecificValues>
      </CacheNodeTypeSpecificParameter>
    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>

```

...output omitted...

```
</ResponseMetadata>  
</DescribeCacheParametersResponse>
```

Para obter mais informações, consulte [DescribeCacheParameters](#).

Modificação de um grupo de parâmetros do ElastiCache

Important

Não é possível modificar um parameter group padrão.

Você pode modificar alguns valores de parâmetros em um parameter group. Esses valores de parâmetros são aplicados a clusters associados ao parameter group. Para obter mais informações sobre quando uma mudança no valor do parâmetro é aplicada a um parameter group, consulte [Parâmetros do Valkey e do Redis OSS](#) e [Parâmetros específicos do Memcached](#).

Modificação de um grupo de parâmetros (console)

O procedimento a seguir mostra como alterar o valor do parâmetro `cluster-enabled` usando o console do ElastiCache. Você usaria o mesmo procedimento para alterar o valor de qualquer parâmetro.

Para alterar o valor de um parâmetro usando o console do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Escolha o parameter group que você deseja modificar, marcando a caixa à esquerda do nome do parameter group.

Os parâmetros do parameter group serão listados na parte inferior da tela. Talvez você precise pular pela lista para ver todos os parâmetros.

4. Para modificar um ou mais parâmetros, escolha Edit Parameters.
5. Na tela Edit Parameter Group:, percorra usando as setas esquerda e direita até encontrar o parâmetro `binding_protocol`. Em seguida, digite `ascii` na coluna Value.
6. Escolha Salvar alterações.

7. Para o Memcached, para localizar o nome do parâmetro que você alterou, consulte [Parâmetros específicos do Memcached](#). Se alterações no parâmetro ocorrerem depois da reinicialização (After restart), reinicialize todos os clusters que usam esse parameter group. Para obter mais informações, consulte [Rebooting clusters](#).
8. Com Valkey e Redis OSS, para localizar o nome do parâmetro que você alterou, consulte [Parâmetros do Valkey e do Redis OSS](#). Se você tiver um cluster do Valkey ou Redis OSS (modo cluster desabilitado) e fizer alterações nos parâmetros a seguir, é preciso reinicializar os nós no cluster:
 - activerehashing
 - bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

 Alterações de parâmetro do Valkey ou Redis OSS (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster do Valkey ou Redis OSS (modo cluster habilitado), siga as etapas a seguir.

- activerehashing
 - bancos de dados
1. Com o Redis OSS, você pode criar um backup manual do cluster. Consulte [Realização de backups manuais](#).
 2. Excluir o cluster. Consulte [Deleting clusters](#).
 3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Modificação de um grupo de parâmetros (AWS CLI)

Para alterar o valor de um parâmetro usando a AWS CLI, use o comando `modify-cache-parameter-group`.

Example

Com o Memcached, para localizar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros específicos do Memcached](#)

O código de exemplo a seguir define o valor de dois parâmetros, `chunk_size` e `chunk_size_growth_fact` no parameter group `myMem14`.

Para 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
```

Para 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
```

A saída desse comando será semelhante a esta.

```
{  
  "CacheParameterGroupName": "myMem14"  
}
```

Example

Com o Valkey e Redis OSS, para localizar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros do Valkey e do Redis OSS](#)

O código de exemplo a seguir define o valor de dois parâmetros, `reserved-memory-percent` e `cluster-enabled` no parameter group `myredis32-on-30`. Definimos `reserved-memory-percent` como 30 (30%) e `cluster-enabled` como `yes` para que o grupo de parâmetros possa ser usado com clusters do Valkey ou Redis OSS (modo cluster habilitado) (grupos de replicação).

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myredis32-on-30 \  
  --parameter-name-values \  
    ParameterName=reserved-memory-percent,ParameterValue=30 \  
    ParameterName=cluster-enabled,ParameterValue=yes
```

Para Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myredis32-on-30 ^  
  --parameter-name-values ^  
    ParameterName=reserved-memory-percent,ParameterValue=30 ^  
    ParameterName=cluster-enabled,ParameterValue=yes
```

A saída desse comando será semelhante a esta.

```
{  
  "CacheParameterGroupName": "my-redis32-on-30"  
}
```

Para obter mais informações, consulte [modify-cache-parameter-group](#).

Para localizar o parâmetro que você alterou, consulte [Parâmetros do Valkey e do Redis OSS](#).

Se você tiver um cluster do Valkey ou Redis OSS (modo cluster desabilitado) e fizer alterações nos parâmetros a seguir, é preciso reinicializar os nós no cluster:

- activerehashing
- bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

Alterações de parâmetro do Valkey ou Redis OSS (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster do Valkey ou Redis OSS (modo cluster habilitado), siga as etapas a seguir.

- activerehashing
- bancos de dados

1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
2. Excluir o cluster. Consulte [Deleting clusters](#).
3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Modificação de um grupo de parâmetros (API do ElastiCache)

Para alterar os valores dos parâmetros de um grupo usando a API do ElastiCache, use a ação `ModifyCacheParameterGroup`.

Example

Com o Memcached, para localizar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros específicos do Memcached](#)

O código de exemplo a seguir define o valor de dois parâmetros, `chunk_size` e `chunk_size_growth_fact` no parameter group `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheParameterGroup
&CacheParameterGroupName=myMem14
&ParameterNameValues.member.1.ParameterName=chunk_size
&ParameterNameValues.member.1.ParameterValue=96
&ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact
&ParameterNameValues.member.2.ParameterValue=1.5
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Example

Com o Valkey e Redis OSS, para localizar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros do Valkey e do Redis OSS](#)

O código de exemplo a seguir define o valor de dois parâmetros, `reserved-memory-percent` e `cluster-enabled` no parameter group `myredis32-on-30`. Definimos `reserved-memory-percent` como 30 (30%) e `cluster-enabled` como `yes` para que o grupo de parâmetros possa ser usado com clusters do Valkey ou Redis OSS (modo cluster habilitado) (grupos de replicação).

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheParameterGroup
&CacheParameterGroupName=myredis32-on-30
&ParameterNameValues.member.1.ParameterName=reserved-memory-percent
&ParameterNameValues.member.1.ParameterValue=30
&ParameterNameValues.member.2.ParameterName=cluster-enabled
&ParameterNameValues.member.2.ParameterValue=yes
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyCacheParameterGroup](#).

Se você tiver um cluster do Valkey ou Redis OSS (modo cluster desabilitado) e fizer alterações nos parâmetros a seguir, é preciso reinicializar os nós no cluster:

- `activeresharding`
- bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

Alterações de parâmetro do Valkey ou Redis OSS (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster do Valkey ou Redis OSS (modo cluster habilitado), siga as etapas a seguir.

- `activeresharding`
- bancos de dados

1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
2. Excluir o cluster. Consulte [Excluindo um cluster no ElastiCache](#).

3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Exclusão de um grupo de parâmetros do ElastiCache

Você pode excluir um grupo de parâmetros personalizado usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Não será possível excluir um parameter group se ele estiver associado a qualquer cluster. Você também não pode excluir nenhum dos parameter groups padrão.

Exclusão de um grupo de parâmetros (console)

O procedimento a seguir mostra como excluir um grupo de parâmetros usando o console do ElastiCache.

Para excluir um grupo de parâmetros usando o console do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Escolha os parameter groups que você deseja excluir marcando a caixa à esquerda do nome do parameter group.

O botão Delete ficará ativo.

4. Escolha Excluir.

A tela de confirmação Delete Parameter Groups será exibida.

5. Para excluir os parameter groups, na tela de confirmação Delete Parameter Groups, escolha Delete.

Para manter os parameter groups, escolha Cancel.

Exclusão de um grupo de parâmetros (AWS CLI)

Para excluir um grupo de parâmetros usando a AWS CLI, use o comando `delete-cache-parameter-group`. Para o parameter group a ser excluído, o parameter group especificado por `--cache-parameter-group-name` não pode ter nenhum cluster associado a ele, nem pode ser um parameter group padrão.

O código de exemplo a seguir exclui o parameter group myMem14.

Example

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myRed28
```

Para Windows:

```
aws elasticache delete-cache-parameter-group ^  
  --cache-parameter-group-name myRed28
```

Para obter mais informações, consulte [delete-cache-parameter-group](#).

Exclusão de um grupo de parâmetros (API do ElastiCache)

Para excluir um grupo de parâmetros usando a API do ElastiCache, use a ação `DeleteCacheParameterGroup`. Para o parameter group a ser excluído, o parameter group especificado por `CacheParameterGroupName` não pode ter nenhum cluster associado a ele, nem pode ser um parameter group padrão.

Example

Com o Memcached, o código de exemplo a seguir exclui o grupo de parâmetros `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myMem14  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

Example

O código de exemplo a seguir exclui o parameter group `myRed28`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myRed28
```

```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DeleteCacheParameterGroup](#).

Parâmetros específicos do mecanismo

Valkey e Redis OSS

A maioria dos parâmetros do Valkey 8 é compatível com os parâmetros do Redis OSS 7.1. Os parâmetros do Valkey 7.2 são iguais aos parâmetros do Redis OSS 7.

Se você não especificar um grupo de parâmetros para seu cluster do Valkey ou Redis OSS, será usado um grupo de parâmetros padrão apropriado à versão de seu mecanismo. Não é possível alterar os valores de nenhum parâmetro em um grupo de parâmetros padrão. No entanto, é possível criar um grupo de parâmetros personalizado e atribuí-lo ao seu cluster a qualquer momento, desde que os valores de parâmetros condicionalmente modificáveis sejam os mesmos nos dois grupos de parâmetros. Para obter mais informações, consulte [Criação de um grupo de parâmetros do ElastiCache](#).

Tópicos

- [Parâmetros do Valkey e do Redis OSS](#)
- [Parâmetros específicos do Memcached](#)

Parâmetros do Valkey e do Redis OSS

Tópicos

- [Alterações de parâmetros do Valkey 8.2](#)
- [Alterações de parâmetros do Valkey 8.1](#)
- [Alterações de parâmetros do Valkey 8.0](#)
- [Alterações nos parâmetros do Valkey 7.2 e Redis OSS 7](#)
- [Alterações de parâmetros do Redis OSS 6.x](#)
- [Alterações de parâmetros do Redis OSS 5.0.3](#)
- [Alterações de parâmetros do Redis OSS 5.0.0](#)
- [Alterações de parâmetros do Redis OSS 4.0.10](#)
- [Alterações de parâmetros do Redis OSS 3.2.10](#)
- [Alterações de parâmetros do Redis OSS 3.2.6](#)
- [Alterações de parâmetros do Redis OSS 3.2.4](#)
- [Parâmetros adicionados do Redis OSS 2.8.24 \(aprimorado\)](#)
- [Parâmetros adicionados do Redis OSS 2.8.23 \(aprimorado\)](#)
- [Parâmetros adicionados do Redis OSS 2.8.22 \(aprimorado\)](#)
- [Parâmetros adicionados do Redis OSS 2.8.21](#)
- [Parâmetros adicionados do Redis OSS 2.8.19](#)
- [Parâmetros adicionados do Redis OSS 2.8.6](#)
- [Parâmetros do Redis OSS 2.6.13](#)
- [Parâmetros específicos de node-type do Redis OSS](#)

Alterações de parâmetros do Valkey 8.2

Família de grupos de parâmetros: valkey8

Note

- As alterações dos parâmetros do Valkey 8.2 não se aplicam ao Valkey 8.1
- Os grupos de parâmetros do Valkey 8.0 e superiores são incompatíveis com o Redis OSS 7.2.4.

- No Valkey 8.2, os seguintes comandos não estão disponíveis para caches de tecnologia sem servidor: `commandlog`, `commandlog get`, `commandlog help`, `commandlog len` e `commandlog reset`.

Novos grupos de parâmetros no Valkey 8.2

Nome	Detalhes	Description
search-fanout-target-mode (adicionado em 8.2)	Padrão: cliente Tipo: string Modificável: sim As alterações entram em vigor: imediatamente	O parâmetro search-fanout-target-mode de configuração controla como as consultas de pesquisa são distribuídas entre os nós em um ambiente de cluster Valkey. Essa configuração aceita dois valores: “throughput”, que otimiza o throughput máximo distribuindo aleatoriamente as consultas de pesquisa em todos os nós do cluster, independentemente do tipo de cliente ou do status READONLY, e “client”, que respeita as características da conexão do cliente ao rotear aleatoriamente clientes não READONLY para nós primários, clientes READONLY em conexões de réplica somente para nós de réplica e clientes READONLY em conexões primárias aleatoriamente em todos os nós.

Nome	Detalhes	Description
		O comportamento padrão é o modo “client”, o que significa que o sistema respeitará os tipos de conexão do cliente e o status READONLY para decisões de roteamento de consultas. Use o modo de taxa de transferência para cargas de trabalho de pesquisa de alto volume, onde a utilização máxima dos recursos do cluster é desejada, e o modo cliente, quando quiser manter a read/write separação e respeitar os padrões de conexão READONLY em nível de aplicativo.
search-default-timeout-ms	Padrão: 50000 Valores permitidos: 1 a 60000 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	O tempo limite padrão de consulta de pesquisa do Valkey (em milissegundos).

Nome	Detalhes	Description
search-enable-partial-results	Padrão: sim Valores permitidos: sim, não Tipo: booleano Modificável: sim As alterações entram em vigor: imediatamente	Configura o comportamento da falha de consulta para a pesquisa do Valkey. Quando ativadas, as consultas de pesquisa retornarão resultados parciais se houver tempos limite em um ou mais fragmentos. Quando desativado, qualquer tempo limite de fragmento fará com que toda a consulta de pesquisa falhe e retorne um erro.

Alterações de parâmetros do Valkey 8.1

Família de grupos de parâmetros: valkey8

Note

- As alterações dos parâmetros do Valkey 8.1 não se aplicam ao Valkey 8.0
- Os grupos de parâmetros do Valkey 8.0 e superiores são incompatíveis com o Redis OSS 7.2.4.
- No Valkey 8.1, os seguintes comandos não estão disponíveis para caches de tecnologia sem servidor: `commandlog`, `commandlog get`, `commandlog help`, `commandlog len` e `commandlog reset`.

Novos grupos de parâmetros no Valkey 8.1

Nome	Detalhes	Description
commandlog-large-request-max-len (adicionado em 8.1)	Padrão: 1048576	O tamanho máximo, em bytes, das solicitações a serem

Nome	Detalhes	Description
	Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	registradas pelo atributo Valkey Command Log.
commandlog-large-request-max-len (adicionado em 8.1)	Padrão: 128 Valores permitidos: 0-1024 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	O comprimento máximo do log de comandos do Valkey para solicitações.
commandlog-reply-larger-than (adicionado em 8.1)	Padrão: 1048576 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	O tamanho máximo, em bytes, das respostas a serem registradas pelo atributo Valkey Command Log.
commandlog-large-reply-max-len (adicionado em 8.1)	Padrão: 128 Valores permitidos: 0-1024 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	O comprimento máximo do log de comandos do Valkey para respostas.

Alterações de parâmetros do Valkey 8.0

Família de grupos de parâmetros: valkey8

Note

O Redis OSS 7.2.4 é incompatível com os grupos de parâmetros Valkey 8 e superiores.

Mudanças de parâmetros específicos no Valkey 8.0

Nome	Detalhes	Description
repl-backlog-size	Padrão: 10485760 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	O tamanho, em bytes, do buffer de backlog do nó primário. O backlog é usado para registrar atualizações de dados no nó primário. Quando uma réplica de leitura se conecta ao primário, ela tenta executar uma sincronização parcial (psync), onde aplica dados do backlog para chegar ao mesmo nível do nó primário. Se psync falhar, será necessária uma sincronização completa. O valor mínimo para esse parâmetro é 16384. Observação: a partir do Redis OSS 2.8.22, esse parâmetro se aplica ao cluster primário, bem como às réplicas de leitura.
maxmemory-samples	Padrão: 3	

Nome	Detalhes	Description
	Valores permitidos: 1 a 64 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	Para cálculos least-recently-used (LRU) e time-to-live (TTL), esse parâmetro representa o tamanho amostral das chaves a serem verificadas. Por padrão, o Redis OSS escolhe três chaves e usa a que foi usada menos recentemente.

Novos grupos de parâmetros no Valkey 8.0

Nome	Detalhes	Description
extended-redis-compatibility	Valores permitidos: sim, não Padrão: sim Tipo: booleano Modificável: sim As alterações ocorrem: imediatamente	O modo de compatibilidade estendida do Redis OSS faz com que o Valkey finja ser o Redis OSS 7.2. Ative essa opção somente se você tiver problemas com ferramentas ou clientes. Impactos para o cliente: • LOADING: o Redis OSS está carregando o conjunto de dados na memória • BUSY: o Redis OSS está ocupado • MISCONF: o Redis OSS é configurado de uma das seguintes formas: • O comando HELLO retorna “server” =>

Nome	Detalhes	Description
		“redis” e “version” => “7.2.4” (nossa versão de compatibilidade com o Redis OSS). O campo INFO para o modo é chamado de “redis_mode”.

Grupos de parâmetros removidos no Valkey 8.0

Nome	Detalhes	Description
lazyfree-lazy-eviction	Valores permitidos: sim, não Padrão: não Tipo: booliano Modificável: sim As alterações ocorrem: imediatamente	Realiza uma exclusão assíncrona em remoções.
lazyfree-lazy-expire	Valores permitidos: sim, não Padrão: não Tipo: booliano Modificável: sim As alterações ocorrem: imediatamente	Realiza uma exclusão assíncrona em chaves expiradas.
lazyfree-lazy-server-del	Valores permitidos: sim, não	

Nome	Detalhes	Description
	Padrão: não Tipo: booleano Modificável: sim As alterações ocorrem: imediatamente	Realiza uma exclusão assíncrona para comandos que atualizam valores.
lazyfree-lazy-user-del	Padrão: não Tipo: string Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Quando o valor é definido como yes (sim), o comando DEL funciona da mesma forma que UNLINK.
replica-lazy-flush	Padrão: sim Tipo: booleano Modificável: não Nome anterior: slave-lazy-flush	Realiza um flushDB assíncrono durante a sincronização da réplica.

Alterações nos parâmetros do Valkey 7.2 e Redis OSS 7

Família de grupos de parâmetros: valkey7

Os grupos de parâmetros padrão do Valkey 7.2 são os seguintes:

- `default.valkey7`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey (modo cluster desabilitado).

- `default.valkey7.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey (modo cluster habilitado).

Família de grupo de parâmetros: redis7

Os grupos de parâmetros padrão do Redis OSS 7 são os seguintes:

- `default.redis7`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Redis OSS (modo cluster desabilitado).
- `default.redis7.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Redis OSS (modo cluster habilitado).

Alterações específicas de parâmetros

Os parâmetros adicionados no Redis OSS 7 são os seguintes. O Valkey 7.2 também aceita esses parâmetros.

Nome	Detalhes	Description
<code>cluster-allow-pubsubshard-when-down</code>	Valores permitidos: yes, no Padrão: yes Tipo: string Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Quando definido como padrão de yes (sim), permite que os nós forneçam tráfego fragmentado do pubsub enquanto o cluster está inativo, desde que acredite que possui os slots.
<code>cluster-preferred-endpoint-type</code>	Valores permitidos: ip, tls-dynamic Padrão: tls-dynamic Tipo: string	Esse valor controla qual endpoint é retornado para solicitações MOVED/ASKING, bem como o campo do endpoint para CLUSTER SLOTS e CLUSTER SHARDS. Quando o valor for definido como ip, o nó anunciará seu endereço IP. Quando o valor é definido como tls-dynamic, o nó anunciará um nome de host quando

Nome	Detalhes	Description
	Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	encryption-in-transit estiver habilitado e, caso contrário, um endereço IP.
latency-tracking	Valores permitidos: yes, no Padrão: no Tipo: string Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Quando definido como yes (sim), rastreia as latências por comando e permite exportar a distribuição de percentil por meio do comando de estatísticas de latência do INFO e as distribuições de latência cumulativa (histogramas) por meio do comando LATENCY.
hash-max-listpack-entries	Valores permitidos: 0+ Padrão: 512 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O número máximo de entradas de hash para que o conjunto de dados seja compactado.

Nome	Detalhes	Description
hash-max-listpack-value	Valores permitidos: 0+ Padrão: 64 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O limite das maiores entradas de hash para que o conjunto de dados seja compactado.
zset-max-listpack-entries	Valores permitidos: 0+ Padrão: 128 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O número máximo de entradas do conjunto classificado para que o conjunto de dados seja compactado.
zset-max-listpack-value	Valores permitidos: 0+ Padrão: 64 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O limite das maiores entradas do conjunto classificado para que o conjunto de dados seja compactado.

Os parâmetros alterados no Redis OSS 7 são os seguintes.

Nome	Detalhes	Description
activerhashing	Permite modificação: no. No Redis OSS 7, esse parâmetro está oculto e ativado por padrão. Para desativá-lo, você precisa criar um caso de suporte .	Permite modificação era Sim.

Os parâmetros removidos no Redis OSS 7 são os seguintes.

Nome	Detalhes	Description
hash-max-ziplist-entries	Valores permitidos: 0+ Padrão: 512 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Use listpack em vez de ziplist para representar uma pequena codificação de hash
hash-max-ziplist-value	Valores permitidos: 0+ Padrão: 64 Tipo: inteiro Modificável: sim	Use listpack em vez de ziplist para representar uma pequena codificação de hash

Nome	Detalhes	Description
	As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	
<code>zset-max-ziplist-entries</code>	Valores permitidos: 0+ Padrão: 128 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Use listpack em vez de ziplist para representar uma pequena codificação de hash.
<code>zset-max-ziplist-value</code>	Valores permitidos: 0+ Padrão: 64 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Use listpack em vez de ziplist para representar uma pequena codificação de hash.

Nome	Detalhes	Description
<code>list-max-ziplist-size</code>	Valores permitidos: Padrão: -2 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O número de entradas permitidas por nó interno da lista.

Alterações de parâmetros do Redis OSS 6.x

Família de grupo de parâmetros: `redis6.x`

Os grupos de parâmetros padrão do Redis OSS 6.x são os seguintes:

- `default.redis6.x`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster desabilitado).
- `default.redis6.x.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster habilitado).

Note

No mecanismo Redis OSS versão 6.2, quando a família de nós `r6gd` foi introduzida para uso com [Classificação de dados em níveis no ElastiCache](#), somente as políticas `max-memory noeviction`, `volatile-lru` e `allkeys-lru` são compatíveis com os tipos de nó `r6gd`.

Para obter mais informações, consulte [ElastiCache versão 6.2 para Redis OSS \(aprimorada\)](#) e [ElastiCache versão 6.0 para Redis OSS \(aprimorada\)](#).

Os parâmetros adicionados no Redis OSS 6.x são os seguintes.

Detalhes	Description	
acl-pubsub-default (added in 6.2)	Valores permitidos: <code>resetchannels</code> , <code>allchannels</code> Padrão: <code>allchannels</code> Tipo: string Modificável: sim Entrada das alterações em vigor: os usuários existentes do Redis OSS associados ao cluster continuarão a ter as permissões existentes. Atualize os usuários ou reinicialize o cluster para atualizar os usuários existentes do Redis OSS.	Permissões de canal pubsub padrão para usuários de ACL implantados neste cluster.
cluster-allow-reads-when-down (added in 6.0)	Padrão: não Tipo: string Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Quando definido como sim, um grupo de replicação do Redis OSS (modo cluster habilitado) continua a processar comandos de leitura mesmo quando um nó não é capaz de alcançar um quórum de primários. Quando definido como o padrão de não (não), o grupo de replicação rejeita todos os comandos. Recomendamos definir esse valor como yes (sim) se você estiver usando um cluster com menos de três grupos de nós ou sua aplicação pode lidar com leituras obsoletas com segurança.

Detalhes	Description	
<code>tracking-table-max-keys</code> (added in 6.0)	Padrão: 1.000.000 Tipo: número Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Para ajudar o cache do lado do cliente, o Redis OSS oferece suporte ao monitoramento de quais clientes acessaram quais chaves. Quando a chave monitorada é modificada, mensagens de invalidação são enviadas a todos os clientes para notificá-los que seus valores armazenados em cache não são mais válidos. Esse valor permite que você especifique o limite superior desta tabela. Depois que esse valor de parâmetro é excedido, são enviadas invalidações aos clientes aleatoriamente. Esse valor deve ser ajustado para limitar o uso da memória, enquanto ainda mantém o controle de chaves suficientes. As chaves também são invalidadas em condições de pouca memória.
<code>acllog-max-len</code> (added in 6.0)	Padrão: 128 Tipo: número Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Esse valor corresponde ao número máximo de entradas no log da ACL.

Detalhes	Description	
active-expire-effort (added in 6.0)	Padrão: 1 Tipo: número Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O Redis OSS exclui chaves que excederam seu tempo de vida por dois mecanismos. Em um, uma chave é acessada e se descobre que ela está expirada. No outro, um trabalho periódico amostra as chaves e faz com que aquelas que excederam seu tempo de vida expirem. Esse parâmetro define a quantidade de esforço que o Redis OSS usa para expirar itens no trabalho periódico. O valor padrão de 1 tenta evitar ter mais de 10 por cento das chaves expiradas ainda na memória. Ele também tenta evitar consumir mais de 25% da memória total e adicionar latência ao sistema. Você pode aumentar esse valor até 10 para aumentar a quantidade e de esforço gasto em chaves expirando. A compensação é mais CPU e latência potencialmente maior. Recomendamos um valor de 1, a menos que você esteja vendo alto uso de memória e possa tolerar um aumento na utilização da CPU.
lazyfree-lazy-user-del (added in 6.0)	Padrão: não Tipo: string Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Quando o valor é definido como yes (sim), o comando DEL funciona da mesma forma que UNLINK.

Os parâmetros removidos no Redis OSS 6.x são os seguintes.

Nome	Detalhes	Description
<code>lua-replicate-commands</code>	Valores permitidos: sim/não Padrão: sim Tipo: booliano Modificável: sim As alterações entram em vigor: imediatamente	Sempre habilite ou não a replicação de efeito Lua em scripts Lua

Alterações de parâmetros do Redis OSS 5.0.3

Família de grupo de parâmetros: `redis5.0`

Grupos de parâmetros padrão do Redis OSS 5.0

- `default.redis5.0`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster desabilitado).
- `default.redis5.0.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster habilitado).

Parâmetros adicionados no Redis OSS 5.0.3

Nome	Detalhes	Description
<code>rename-commands</code>	Padrão: nenhum Tipo: string Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Uma lista separada por espaços de comandos Redis OSS renomeados. Veja a seguir uma lista restrita de comandos disponíveis para renomear: APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLPUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR DECRBY DEL DISCARD DUMP ECHO EVAL

Nome	Detalhes	Description
		EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS GEORADIUSBYMEMBER GET GETBIT GETRANGE GETSET HDEL HEXISTS HGET HGETALL HINCRBY HINCRBYFL OAT HKEYS HLEN HMGET HSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPU SH LPUSHX LRANGE LREM LSET LTRIM MEMORY MGET MONITOR MOVE MSET MSETNX MULTI OBJECT PERSIST PEXPIRE PEXPIREAT PFADD PFCOUNT PFMERGE PING PSETEX PSUBSCRIBE PUBSUB PTTL PUBLISH PUNSUBSCRIBE RANDOMKEY READONLY READWRITE RENAME RENAMENX RESTORE ROLE RPOP RPOPLPUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET SETBIT SETEX SETNX SETRANGE SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP SRANDMEMBER SREM STRLEN SUBSCRIBE UNION UNIONSTORE SWAPDB TIME TOUCH TTL TYPE UNSUBSCRIBE UNLINK UNWATCH WAIT WATCH ZADD ZCARD ZCOUNT ZINCRBY ZINTERSTO RE ZLEXCOUNT ZPOPMAX ZPOPMIN ZRANGE ZRANGEBYLEX ZREVRANGE BYLEX ZRANGEBYSCORE ZRANK ZREM ZREMRANGEBYLEX ZREMRANGEBYRANK ZREMRANGEBYSCORE ZREVRANGE ZREVRANGEBYSCORE ZREVRANK ZSCORE

Nome	Detalhes	Description
		ZUNIONSTORE SCAN SSCAN HSCAN ZSCAN XINFO XADD XTRIM XDEL XRA NGE XREVRANGE XLEN XREAD XGROUP XREADGROUP XACK XCLAIM XPENDING GEORADIUS_RO GEORADIUSBYMEMBER_ RO LOLWUT XSETID SUBSTR

Para obter mais informações, consulte [ElastiCache versão 5.0.6 para Redis OSS \(aprimorada\)](#).

Alterações de parâmetros do Redis OSS 5.0.0

Família de grupo de parâmetros: redis5.0

Grupos de parâmetros padrão do Redis OSS 5.0

- `default.redis5.0`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster desabilitado).
- `default.redis5.0.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster habilitado).

Parâmetros adicionados no Redis OSS 5.0

Nome	Detalhes	Description
<code>stream-node-max-bytes</code>	Valores permitidos: 0+ Padrão: 4096 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	A estrutura do fluxo de dados é uma árvore radix de nós que codifica vários itens dentro. Use esta configuração para especificar o tamanho máximo de um nó único em uma árvore radix em bytes. Se definido como 0, o tamanho do nó da árvore é ilimitado.

Nome	Detalhes	Description
<code>stream-no-de-max-entries</code>	Valores permitidos: 0+ Padrão: 100 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	A estrutura do fluxo de dados é uma árvore radix de nós que codifica vários itens dentro. Use essa configuração para especificar o número máximo de itens que um único nó pode conter antes de alternar para um novo nó ao anexar novas entradas de fluxo. Se definido como 0, o número de itens no nó da árvore é ilimitado
<code>active-defrag-max-scan-fields</code>	Valores permitidos: 1 a 1000000 Padrão: 1000 Tipo: inteiro Modificável: sim As alterações entram em vigor: imediatamente	Número máximo de set/hash/zset/list campos que serão processados a partir da verificação do dicionário principal
<code>lua-replique-commands</code>	Valores permitidos: sim/não Padrão: sim Tipo: booleano Modificável: sim As alterações entram em vigor: imediatamente	Sempre habilite ou não a replicação de efeito Lua em scripts Lua

Nome	Detalhes	Description
<code>replica-ignore-maxmemory</code>	Padrão: sim Tipo: booliano Modificável: não	Determina se a réplica ignora a configuração <code>maxmemory</code> ao não remover itens independentes do primário

O Redis OSS renomeou vários parâmetros na versão 5.0 do mecanismo em resposta ao feedback da comunidade. Para obter mais informações, consulte [O que há de novo no Redis OSS 5?](#) A tabela a seguir lista os novos nomes e como eles mapeiam para versões anteriores.

Parâmetros renomeados no Redis OSS 5.0

Nome	Detalhes	Description
<code>replica-lazy-flush</code>	Padrão: sim Tipo: booliano Modificável: não Nome anterior: <code>slave-lazy-flush</code>	Realiza um flushDB assíncrono durante a sincronização da réplica.
<code>client-output-buffer-limit-replica-hard-limit</code>	Padrão: para valores, consulte Parâmetros específicos de node-type do Redis OSS Tipo: inteiro Modificável: não Nome anterior: <code>client-output-buffer-limit - slave-hard-limit</code>	Para réplicas de leitura do Redis OSS: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado.
<code>client-output-buffer-limit-</code>	Padrão: para valores, consulte Parâmetros específicos de node-type do Redis OSS	Para réplicas de leitura do Redis OSS: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado.

Nome	Detalhes	Description
<code>replica-soft-limit</code>	Tipo: inteiro Modificável: não Nome anterior: <code>client-output-buffer-limit - slave-soft-limit</code>	ado, mas somente se essa condição persistir por <code>client-output-buffer-limit-replica-soft-seconds</code> .
<code>client-output-buffer-limit-replica-soft-seconds</code>	Padrão: 60 Tipo: inteiro Modificável: não Nome anterior: <code>client-output-buffer-limit - slave-soft-seconds</code>	Para réplicas de leitura do Redis OSS: se o buffer de saída de um cliente permanecer em <code>client-output-buffer-limit-replica-soft-limit</code> bytes por mais tempo que esse número de segundos, o cliente será desconectado.
<code>replica-allow-chaining</code>	Padrão: não Tipo: string Modificável: não Nome anterior: <code>slave-allow-chaining</code>	Determina se uma réplica de leitura no Redis OSS pode ter réplicas de leitura próprias.
<code>min-replicas-to-write</code>	Padrão: 0 Tipo: inteiro Modificável: sim Nome anterior: <code>min-slaves-to-write</code> As alterações terão efeito: imediatamente	O número mínimo de réplicas de leitura que devem estar disponíveis para que o nó primário aceite gravações de clientes. Se o número de réplicas disponíveis cair abaixo desse número, o nó primário não aceitará mais solicitações de gravação. Se esse parâmetro ou <code>min-replicas-max-lag</code> for 0, o nó primário sempre aceitará solicitações de gravação, mesmo que nenhuma réplica esteja disponível.

Nome	Detalhes	Description
<code>min-replicas-max-lag</code>	Padrão: 10 Tipo: inteiro Modificável: sim Nome anterior: min-slaves-max-lag As alterações terão efeito: imediatamente	O número de segundos em que o nó primário deve receber uma solicitação de ping de uma réplica de leitura. Se essa quantidade de tempo passar, e o primário não receber um ping, a réplica não será mais considerada disponível. Se o número de réplicas disponíveis cair abaixo min-replicas-to-write, a primária deixará de aceitar gravações nesse momento. Se esse parâmetro ou min-replicas-to-write for 0, o nó primário sempre aceitará solicitações de gravação, mesmo que nenhuma réplica esteja disponível.
<code>close-on-replica-write</code>	Padrão: sim Tipo: booliano Modificável: sim Nome anterior: close-on-slave-write As alterações terão efeito: imediatamente	Se ativado, os clientes que tentarem gravar em uma réplica somente leitura serão desconectados.

Parâmetros removidos no Redis OSS 5.0

Nome	Detalhes	Description
<code>repl-timeout</code>	Padrão: 60 Modificável: não	O parâmetro não está disponível nesta versão.

Alterações de parâmetros do Redis OSS 4.0.10

Família de grupo de parâmetros: redis4.0

Grupos de parâmetros padrão do Redis OSS 4.0.x

- `default.redis4.0`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster desabilitado).
- `default.redis4.0.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para grupos de replicação e clusters do Valkey ou Redis OSS (modo cluster habilitado).

Parâmetros alterados no Redis OSS 4.0.10

Nome	Detalhes	Description
maxmemory-policy	Valores permitidos: allkeys-lru , volatile-lru , allkeys-lfu , volatile-lfu , allkeys-random , volatile-random , volatile-ttl , noeviction Padrão: volatile-lru Tipo: string Modificável: sim As alterações ocorrem: imediatamente	maxmemory-policy foi adicionado na versão 2.6.13. Na versão 4.0.10 foram adicionados dois novos valores permitidos: allkeys-lfu , que excluirá qualquer chave que esteja usando LFU aproximado, e volatile-lfu , que excluirá o uso de LFU aproximado entre as chaves com um conjunto de expiração. Na versão 6.2, quando a família de nós r6gd foi introduzida para uso com classificação de dados em níveis, somente as políticas noeviction , volatile-lru e allkeys-lru são compatíveis com os tipos de nó r6gd.

Parâmetros adicionados no Redis OSS 4.0.10

Nome	Detalhes	Description

Parâmetros de exclusão assíncrona

Nome	Detalhes	Description
lazyfree-lazy- eviction	Valores permitidos: sim/não	Realiza uma exclusão assíncrona em remoções.
	Padrão: não	
	Tipo: booliano	
	Modificável: sim	
	As alterações ocorrem: imediatamente	
lazyfree-lazy-expire	Valores permitidos: sim/não	Realiza uma exclusão assíncrona em chaves expiradas.
	Padrão: não	
	Tipo: booliano	
	Modificável: sim	
	As alterações ocorrem: imediatamente	
lazyfree-lazy-server-del	Valores permitidos: sim/não	Realiza uma exclusão assíncrona para comandos que atualizam valores.
	Padrão: não	
	Tipo: booliano	
	Modificável: sim	
	As alterações ocorrem: imediatamente	

Nome	Detalhes	Description
<code>slave-lazy-flush</code>	Valores permitidos: N/D Padrão: não Tipo: booliano Modificável: não As alterações ocorrem: N/D	Realiza um flushDB assíncrono durante a sincronização subordinada.
Parâmetros de LFU		
<code>lfu-log-factor</code>	Valores permitidos: qualquer inteiro > 0 Padrão: 10 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Defina o fator de log, que determina o número de hits de chave para saturar o contador de chaves.
<code>lfu-decay-time</code>	Valores permitidos: qualquer inteiro Padrão: 1 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	O tempo em minutos para diminuir o contador de chaves.
Parâmetros de desfragmentação ativa		

Nome	Detalhes	Description
activedefrag	Valores permitidos: sim/não Padrão: não Tipo: booleano Modificável: sim As alterações ocorrem: imediatamente	Habilita a desfragmentação ativa.  Note Nas versões 7.0 e superiores do Valkey e Redis OSS,AWS pode executar automaticamente a desfragmentação quando operacionalmente necessário, independentemente dessa configuração.
active-defrag-ignore-bytes	Valores permitidos: 10485760-104857600 Padrão: 104857600 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Quantidade mínima de desperdício de fragmentação para iniciar a desfragmentação ativa.

Nome	Detalhes	Description
active-defrag-threshold-lower	Valores permitidos: 1-100 Padrão: 10 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Porcentagem mínima de fragmentação para iniciar a desfragmentação ativa.
active-defrag-threshold-upper	Valores permitidos: 1-100 Padrão: 100 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Porcentagem máxima de fragmentação em que usamos o esforço máximo.
active-defrag-cycle-min	Valores permitidos: 1-75 Padrão: 25 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Esforço mínimo para desfragmentação em porcentagem de CPU.

Nome	Detalhes	Description
active-defrag-cycle-max	Valores permitidos: 1-75 Padrão: 75 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Esforço máximo para desfragmentação em porcentagem de CPU.
Parâmetros de buffer de saída do cliente		
client-query-buffer-limit	Valores permitidos: 1048576-1073741824 Padrão: 1073741824 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Tamanho máximo de um único buffer de consulta do cliente.
proto-max-bulk-len	Valores permitidos: 1048576-536870912 Padrão: 536870912 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Tamanho máximo de uma única solicitação de elemento.

Alterações de parâmetros do Redis OSS 3.2.10

Família de grupo de parâmetros: redis3.2

ElastiCache para o Redis OSS 3.2.10, não há suporte a parâmetros adicionais.

Alterações de parâmetros do Redis OSS 3.2.6

Família de grupo de parâmetros: redis3.2

Para o Redis OSS 3.2.6, não há parâmetros adicionais com suporte.

Alterações de parâmetros do Redis OSS 3.2.4

Família de grupo de parâmetros: redis3.2

Começando com o Redis OSS 3.2.4, existem dois grupo de parâmetros padrão.

- `default.redis3.2`: ao executar o Redis OSS 3.2.4, especifique este grupo de parâmetros, ou um derivado dele, se quiser criar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) e ainda usar os atributos adicionais do Redis OSS 3.2.4.
- `default.redis3.2.cluster.on`: especifique este grupo de parâmetros, ou um derivado dele, quando quiser criar um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado).

Tópicos

- [Novos parâmetros para o Redis OSS 3.2.4](#)
- [Parâmetros alterados no Redis OSS 3.2.4 \(aprimorado\)](#)

Novos parâmetros para o Redis OSS 3.2.4

Família de grupo de parâmetros: redis3.2

Para o Redis OSS 3.2.4, há suporte para os seguintes parâmetros adicionais.

Nome	Detalhes	Description
<code>list-max-ziplist-size</code>	Padrão: -2 Tipo: inteiro	As listas são codificadas de forma especial para economizar espaço. O número de entradas permitidas por nó de lista interno pode ser especificado como um tamanho máximo

Nome	Detalhes	Description
	Modificável: não	fixo ou um número máximo de elementos. Para um tamanho máximo fixo, use de -5 a -1, significando: • -5: tamanho máximo: 64 Kb - não recomendado para workloads normais • -4: tamanho máximo: 32 Kb - não recomendado • -3: tamanho máximo: 16 Kb - não recomendado • -2: tamanho máximo: 8 Kb - recomendado • -1: tamanho máximo: 4 Kb - recomendado • Números positivos significam armazenar exatamente o número de elementos por nó de lista.

Nome	Detalhes	Description
<code>list-compress-depth</code>	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	As listas também podem ser compactadas. A profundidade de compactação é o número de nós ziplist de lista rápida de cada lado da lista a serem excluídos da compactação. O início e o final cauda da lista são sempre descompactados para operações Push e Pop rápidas. As configurações são: • 0: desabilitar toda a compactação. • 1: começar a compactar com o 1º nó de início e final. [início] -> nó-> nó -> ...-> nó -> [final] Todos os nós, exceto [início] e [final] são compactados. • 2: começar a compactar com o 2º nó de início e final. [início] -> [próximo] -> nó-> nó -> ...-> nó -> [anterior] -> [final] [início], [próximo], [anterior], [final] não são compactados. Todos os outros nós são compactados. • Etc.

Nome	Detalhes	Description
<code>cluster-enabled</code>	Padrão: no/yes * Tipo: string Modificável: não	Indica se este é um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) no modo cluster (sim) ou um grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado) no modo sem cluster (não). Grupos de replicação do Valkey ou Redis OSS (modo cluster habilitado) no modo cluster podem particionar seus dados em até 500 grupos de nós. * O Redis OSS 3.2.x tem dois grupo de parâmetros padrão. • <code>default.redis3.2</code> – valor padrão no. • <code>default.redis3.2.cluster.on</code> – valor padrão yes. •

Nome	Detalhes	Description
<code>cluster-require-full-coverage</code>	Padrão: não Tipo: booliano Modificável: sim As alterações terão efeito: imediatamente	Quando estão definidos como yes, os nós do Valkey ou Redis OSS (modo cluster habilitado) no modo cluster param de aceitar consultas quando detectam pelo menos um slot de hash descoberto (nenhum nó disponível que o sirva). Dessa forma, se o cluster estiver parcialmente inativo, ele ficará indisponível. Ele voltará a ficar disponível automaticamente assim que todos os slots forem novamente cobertos. No entanto, às vezes você deseja que o subconjunto do cluster que está trabalhando continue a aceitar consultas para a parte do espaço de chaves que ainda está coberto. Para fazer isto, basta configurar a opção <code>cluster-require-full-coverage</code> como no.

Nome	Detalhes	Description
hll-spars e-max-byt es	Padrão: 3000 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	HyperLogLog limite de bytes de represent ação esparsa. O limite inclui o cabeçalho de 16 bytes. Quando o HyperLogLog uso da representação esparsa ultrapassa esse limite, ele é convertido na representação densa. Não é recomendado um valor superior a 16000, porque, nesse ponto, a representação densa é mais eficiente em termos de memória. Recomendamos um valor de 3000 para ter os benefícios da codificação eficiente do espaço sem diminuir demais o PFADD, que é $O(N)$ com a codificação esparsa. O valor pode ser aumentado para ~ 10000 quando a CPU não é uma preocupação, mas o espaço é, e o conjunto de dados é composto por muitos HyperLogLogs com cardinalidade na faixa de 0 a 15000.

Nome	Detalhes	Description
<code>reserved-memory-percent</code>	Padrão: 25 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	A porcentagem da memória de um nó reservada para o uso não de dados. Por padrão, a área de dados do Redis OSS cresce até consumir toda a memória do nó. Se isso ocorrer, o desempenho do nó provavelmente sofrerá devido a uma paginação de memória excessiva. Ao reservar memória, você pode separar uma parte da memória disponível para fins não Redis OSS, a fim de ajudar a reduzir a quantidade de paginação. Esse parâmetro é específico e não faz parte da distribuição padrão do Redis OSS. ElastiCache Para obter mais informações, consulte <code>reserved-memory</code> e Gerenciamento de memória reservada para Valkey e Redis OSS.

Parâmetros alterados no Redis OSS 3.2.4 (aprimorado)

Família de grupo de parâmetros: redis3.2

Para o Redis OSS 3.2.4, os parâmetros a seguir foram alterados.

Nome	Detalhes	Alteração
<code>activerhashing</code>	Modificável: Sim se o grupo de parâmetros não estiver associado a nenhum cluster. Do contrário, não.	Modificável era Não.
<code>databases</code>	Modificável: Sim se o grupo de parâmetros não estiver associado a nenhum cluster. Do contrário, não.	Modificável era Não.

Nome	Detalhes	Alteração
appendonly	Padrão: desativado Modificável: não	Se quiser atualizar de uma versão anterior do Redis OSS, primeiro você deve desativar appendonly .
appendfsync	Padrão: desativado Modificável: não	Se quiser atualizar de uma versão anterior do Redis OSS, primeiro você deve desativar appendfsync .
repl-timeout	Padrão: 60 Modificável: não	Agora, ele não é modificável com um padrão de 60.
tcp-keepalive	Padrão: 300	O padrão era 0.
list-max-ziplist-entries		O parâmetro não está mais disponível.
list-max-ziplist-value		O parâmetro não está mais disponível.

Parâmetros adicionados do Redis OSS 2.8.24 (aprimorado)

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.24, não há parâmetros adicionais com suporte.

Parâmetros adicionados do Redis OSS 2.8.23 (aprimorado)

Família de grupo de parâmetros: redis2.8

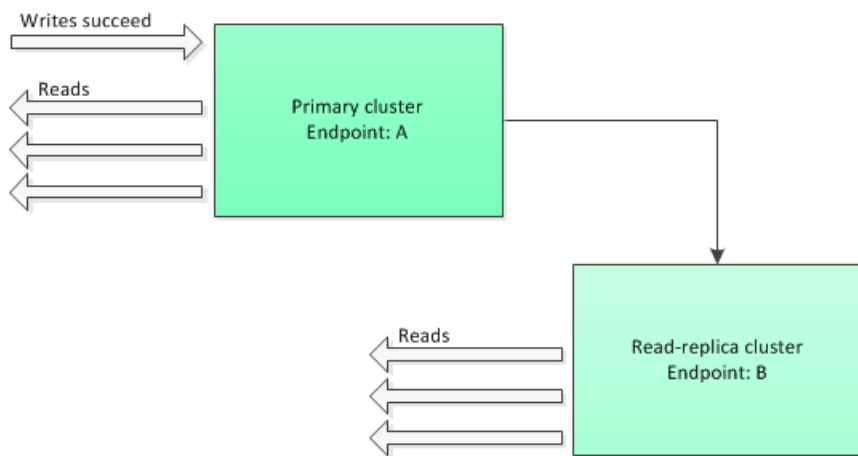
Para o Redis OSS 2.8.23, o parâmetro adicional a seguir tem suporte.

Nome	Detalhes	Description
close-on-slave-write	Padrão: sim Tipo: string (sim/não) Modificável: sim As alterações terão efeito: imediatamente	Se ativado, os clientes que tentarem gravar em uma réplica somente leitura serão desconectados.

Como a close-on-slave-write funciona

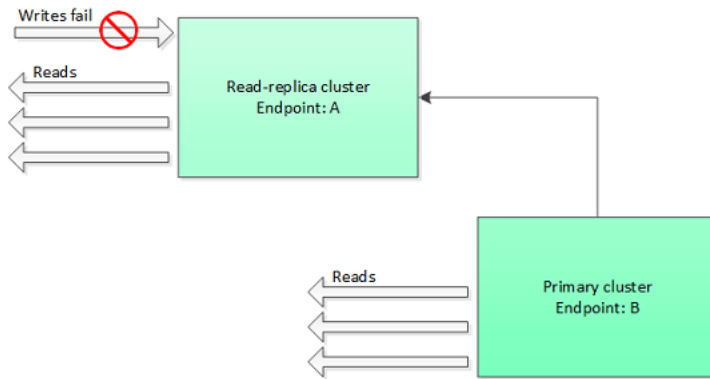
O `close-on-slave-write` parâmetro é introduzido pela Amazon ElastiCache para dar a você mais controle sobre como seu cluster responde quando um nó primário e um nó de réplica de leitura trocam de funções devido à promoção de uma réplica de leitura para primária.

Before read-replica promotion



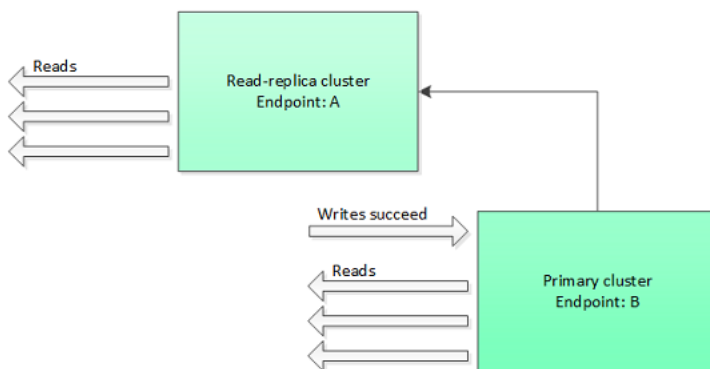
Se o cluster de réplica de leitura for promovido para primário por qualquer motivo que não seja o failover de um grupo de replicação habilitado para Multi-AZ, o cliente continuará tentando gravar no endpoint A. Como o endpoint A é agora o endpoint de uma réplica de leitura, essas gravações falharão. Esse é o comportamento do Redis OSS antes da ElastiCache introdução `close-on-replica-write` e o comportamento se você desabilitar `close-on-replica-write`.

Read-replica Promoted – writes to old primary fail



Com `close-on-replica-write` habilitado, a qualquer momento que um cliente tentar gravar em uma réplica de leitura, a conexão do cliente com o cluster será encerrada. Sua lógica de aplicativo deve detectar a desconexão, verificar a tabela DNS e reconectar-se ao endpoint primário, que agora seria o endpoint B.

Client reconnected to new primary



Quando você pode desativar `close-on-replica-write`

Se desabilitar `close-on-replica-write` resulta na falha do cluster, por que você deseja desabilitar `close-on-replica-write`?

Conforme mencionado anteriormente, com `close-on-replica-write` habilitado, a qualquer momento que um cliente tentar gravar em uma réplica de leitura, a conexão do cliente com o cluster será encerrada. Estabelecer uma nova conexão com o nó leva tempo. Assim, desconectar e reconectar devido a uma solicitação de gravação para a réplica também afeta a latência das solicitações de leitura que são veiculadas pela mesma conexão. Esse efeito permanece em vigor até uma nova conexão ser estabelecida. Se seu aplicativo for especialmente pesado em termos de leitura ou muito sensível à latência, convém manter seus clientes conectados para evitar degradar o desempenho de leitura.

Parâmetros adicionados do Redis OSS 2.8.22 (aprimorado)

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.22, não há parâmetros adicionais com suporte.

Important

- A partir do Redis OSS versão 2.8.22, `repl-backlog-size` aplica-se ao cluster primário, bem como a clusters de réplica.
- A partir do Redis OSS versão 2.8.22, o parâmetro `repl-timeout` não tem suporte. Se for alterado, ElastiCache substituirá pelo padrão (60s), como fazemos com `appendonly`.

Os seguintes parâmetros não têm mais suporte.

- `appendonly`
- `appendfsync`
- `repl-timeout`

Parâmetros adicionados do Redis OSS 2.8.21

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.21, não há parâmetros adicionais com suporte.

Parâmetros adicionados do Redis OSS 2.8.19

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.19, não há parâmetros adicionais com suporte.

Parâmetros adicionados do Redis OSS 2.8.6

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.6, há suporte para os seguintes parâmetros adicionais.

Nome	Detalhes	Description
<code>min-slaves-max-lag</code>	Padrão: 10 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O número de segundos em que o nó primário deve receber uma solicitação de ping de uma réplica de leitura. Se essa quantidade de tempo passar, e o primário não receber um ping, a réplica não será mais considerada disponível. Se o número de réplicas disponíveis cair abaixo <code>min-slaves-to-write</code> e, a primária deixará de aceitar gravações nesse momento. Se esse parâmetro ou <code>min-slaves-to-write</code> for 0, o nó primário sempre aceitará solicitações de gravação, mesmo que nenhuma réplica esteja disponível.
<code>min-slaves-to-write</code>	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O número mínimo de réplicas de leitura que devem estar disponíveis para que o nó primário aceite gravações de clientes. Se o número de réplicas disponíveis cair abaixo desse número, o nó primário não aceitará mais solicitações de gravação. Se esse parâmetro ou <code>min-slaves-max-lag</code> for 0, o nó primário sempre aceitará solicitações de gravação, mesmo que nenhuma réplica esteja disponível.
<code>notify-keyspace-events</code>	Padrão: (uma string vazia)	Os tipos de eventos de espaço de chaves sobre os quais o Redis

Nome	Detalhes	Description
	Tipo: string	OSS pode notificar os clientes.
	Modificável: sim	Cada tipo de evento é representado por uma única letra:
	As alterações terão efeito: imediatamente	• K: eventos de espaço de chaves, publicados com um prefixo de __keyspace@<db>__ • E: eventos-chave, publicados com um prefixo de __keyevent@<db>__ • g — Comandos genéricos não específicos, como DEL, EXPIRE, RENAME, etc. • \$ — Comandos de string • l — Listar comandos • s — Definir comandos • h — Comandos de hash • z — Comandos de conjuntos classificados • x — Eventos expirados (eventos gerados sempre que uma chave expira) • e — Eventos removidos (eventos gerados quando uma chave é removida para maxmemory)

Nome	Detalhes	Description
		• A — Um alias para g\$shzxe Você pode ter qualquer combinação desses tipos de eventos. Por exemplo, AKE significa que o Redis OSS pode publicar notificações de todos os tipos de eventos. Não use nenhum caractere diferente dos listados acima. Se isso for feito, mensagens de erro serão geradas. Por padrão, esse parâmetro é definido como uma string vazia, o que significa que a notificação de eventos de espaço de chaves está desabilitada.

Nome	Detalhes	Description
repl-backlog-size	Padrão: 1048576 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O tamanho, em bytes, do buffer de backlog do nó primário. O backlog é usado para registrar atualizações de dados no nó primário. Quando uma réplica de leitura se conecta ao primário, ela tenta executar uma sincronização parcial (psync), onde aplica dados do backlog para chegar ao mesmo nível do nó primário. Se psync falhar, será necessária uma sincronização completa. O valor mínimo para esse parâmetro é 16384.  Note A partir do Redis OSS 2.8.22, esse parâmetro se aplica ao cluster primário, bem como às réplicas de leitura.

Nome	Detalhes	Description
<code>repl-backlog-ttl</code>	Padrão: 3600 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O número de segundos que o nó primário manterá o buffer de backlog. A partir do momento em que o último nó de réplica tiver sido desconectado, os dados no backlog permanecerão intactos até <code>repl-backlog-ttl</code> expirar. Se a réplica não tiver se conectado ao primário dentro desse tempo, o primário liberará o buffer de backlog. Quando a réplica eventualmente se reconectar, ela terá que executar uma sincronização completa com o primário. Se esse parâmetro estiver definido como 0, o buffer de backlog nunca será liberado.
<code>repl-timeout</code>	Padrão: 60 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Representa o período de tempo limite, em segundos, para: • Transferência de dados em massa durante a sincronização, sob a perspectiva da réplica de leitura • Tempo limite do nó primário, sob a perspectiva da réplica • Tempo limite da réplica, sob a perspectiva do nó primário

Parâmetros do Redis OSS 2.6.13

Família de grupo de parâmetros: redis2.6

O Redis OSS 2.6.13 foi a primeira versão do Redis OSS suportada pelo ElastiCache. A tabela a seguir mostra os parâmetros do Redis OSS 2.6.13 compatíveis. ElastiCache

Nome	Detalhes	Description
activeresharding	Padrão: sim Tipo: string (sim/não) Modificável: sim As alterações ocorrem: na criação	Determina se deseja o recurso de rehashing ativo do Redis deve ou não ser habilitado. A tabela de hash principal é sofre rehashing dez vezes por segundo. Cada operação de rehash consome 1 milissegundo de tempo da CPU. Esse valor é definido quando você cria o grupo de parâmetros. Ao atribuir um novo grupo de parâmetros a um cluster, esse valor deve ser o mesmo nos grupo de parâmetros antigo e novo.
appendonly	Padrão: não Tipo: string Modificável: sim As alterações terão efeito: imediatamente	Habilita ou desabilita o recurso de arquivo somente de acréscimo (AOF) do Redis. O AOF captura todos os comandos do Redis OSS que alteram dados no cache e é usado para recuperação de determinadas falhas de nós. O valor padrão é no, o que significa que o AOF está desligado. Defina esse parâmetro como yes para habilitar o AOF. Para obter mais informações, consulte Atenuar falhas.  Note O recurso Arquivos somente de acréscimo (AOF) não tem suporte para nós cache.t1.micro e cache.t2.*. Para

Nome	Detalhes	Description
		nós desse tipo, o valor do parâmetro <code>appendonly</code> é ignorado.  Note Para grupos de replicação Multi-AZ, o AOF não é permitido.
<code>appendfsync</code>	Padrão: <code>everysec</code> Tipo: <code>string</code> Modificável: <code>sim</code> As alterações terão efeito: imediatamente	Quando <code>appendonly</code> é definido como <code>sim</code>, controla com que frequência o buffer de saída do AOF é gravado no disco: <code>no</code> — o buffer é liberado no disco, conforme necessário. <code>everysec</code>: o buffer é liberado uma vez por segundo. Esse é o padrão. <code>always</code> — o buffer é liberado sempre que os dados no cluster são modificados. - O <code>Appendfsync</code> não é compatível com as versões 2.8.22 e posteriores.
<code>client-output-buffer-limit-normal-hard-limit</code>	Padrão: <code>0</code> Tipo: <code>inteiro</code> Modificável: <code>sim</code> As alterações terão efeito: imediatamente	Se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado. O padrão é zero (sem limite fixo).

Nome	Detalhes	Description
<code>client-output-buffer-limit-normal-soft-limit</code>	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado, mas somente se essa condição persistir por <code>client-output-buffer-limit-normal-soft-seconds</code> . O padrão é zero (sem limite flexível).
<code>client-output-buffer-limit-normal-soft-seconds</code>	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Se o buffer de saída de um cliente permanecer em <code>client-output-buffer-limit-normal-soft-limit</code> bytes por mais tempo que esse número de segundos, o cliente será desconectado. O padrão é zero (sem limite de tempo).
<code>client-output-buffer-limit-pubsub-hard-limit</code>	Padrão: 33554432 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Para publish/subscribe clientes Redis OSS: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado.
<code>client-output-buffer-limit-pubsub-soft-limit</code>	Padrão: 8388608 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Para publish/subscribe clientes Redis OSS: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado, mas somente se essa condição persistir. <code>client-output-buffer-limit-pubsub-soft-seconds</code>

Nome	Detalhes	Description
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	Padrão: 60 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Para publish/subscribe clientes Redis OSS: se o buffer de saída de um cliente permanecer em <code>client-output-buffer-limit-pubsub-soft-limit</code> bytes por mais do que esse número de segundos, o cliente será desconectado.
<code>client-output-buffer-limit-slave-hard-limit</code>	Padrão: para valores, consulte Parâmetros específicos de node-type do Redis OSS Tipo: inteiro Modificável: não	Para réplicas de leitura do Redis OSS: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado.
<code>client-output-buffer-limit-slave-soft-limit</code>	Padrão: para valores, consulte Parâmetros específicos de node-type do Redis OSS Tipo: inteiro Modificável: não	Para réplicas de leitura do Redis OSS: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado, mas somente se essa condição persistir por <code>client-output-buffer-limit-slave-soft-seconds</code> .
<code>client-output-buffer-limit-slave-soft-seconds</code>	Padrão: 60 Tipo: inteiro Modificável: não	Para réplicas de leitura do Redis OSS: se o buffer de saída de um cliente permanecer em <code>client-output-buffer-limit-slave-soft-limit</code> bytes por mais tempo que esse número de segundos, o cliente será desconectado.

Nome	Detalhes	Description
databases	Padrão: 16 Tipo: inteiro Modificável: não As alterações ocorrem: na criação	O número de partições lógicas em que os bancos de dados estão divididos. Recomendamos manter esse valor baixo. Esse valor é definido quando você cria o grupo de parâmetros. Ao atribuir um novo grupo de parâmetros a um cluster, esse valor deve ser o mesmo nos grupo de parâmetros antigo e novo.
hash-max-ziplist-entries	Padrão: 512 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória usada para hashes. Hashes com menos que o número especificado de entradas são armazenados usando uma codificação especial que economiza espaço.
hash-max-ziplist-value	Padrão: 64 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória usada para hashes. Hashes com entradas menores que o número especificado de bytes são armazenados usando uma codificação especial que economiza espaço.
list-max-ziplist-entries	Padrão: 512 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória usada para listas. Listas com menos que o número especificado de entradas são armazenadas usando uma codificação especial que economiza espaço.

Nome	Detalhes	Description
<code>list-max-ziplist-value</code>	Padrão: 64 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória usada para listas. Listas com entradas menores que o número especificado de bytes são armazenadas usando uma codificação especial que economiza espaço.
<code>lua-time-limit</code>	Padrão: 5000 Tipo: inteiro Modificável: não	O tempo máximo de execução de um script Lua, em milissegundos, antes de ElastiCache agir para interromper o script. Se <code>lua-time-limit</code> for excedido, todos os comandos do Redis OSS retornarão um erro no formato <code>____-BUSY</code>. Como esse estado pode causar interferência em muitas operações essenciais do Redis OSS, primeiro ElastiCache emitirá um comando <code>SCRIPT KILL</code>. Se isso não for bem-sucedido, ElastiCache reiniciará forçosamente o Redis OSS.
<code>maxclients</code> Esse valor se aplica a todos os tipos de instâncias, exceto às instâncias explicitamente especificadas	Padrão: 65000 Tipo: inteiro Modificável: não t2.medium Padrão: 20000 Tipo: inteiro Modificável: não t2.small Padrão: 20000 Tipo: inteiro Modificável: não	O número máximo de clientes que podem ser conectados ao mesmo tempo.

Nome	Detalhes	Description
	t2.micro Padrão: 20000	
	Tipo: inteiro	
	Modificável: não	
	t4g.micro Padrão: 20000	
	Tipo: inteiro	
	Modificável: não	
	t3.medium Padrão: 46000	
	Tipo: inteiro	
	Modificável: não	
	t3.small Padrão: 46000	
	Tipo: inteiro	
	Modificável: não	
	t3.micro Padrão: 20000	
	Tipo: inteiro	
	Modificável: não	
maxmemory-policy	Padrão: volatile-lru Tipo: string Modificável: sim As alterações terão efeito: imediatamente	A política de remoção de chaves quando o uso máximo da memória é atingido. Os valores válidos são: volatile-lru allkeys-lru volatile-random allkeys-random volatile-ttl noeviction Para obter mais informações, consulte Usar o Valkey ou Redis OSS como um cache LRU.

Nome	Detalhes	Description
<code>maxmemory-samples</code>	Padrão: 3 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Para cálculos least-recently-used (LRU) e time-to-live (TTL), esse parâmetro representa o tamanho amostral das chaves a serem verificadas. Por padrão, o Redis OSS escolhe 3 chaves e usa a que foi usada menos recentemente.
<code>reserved-memory</code>	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	A memória total, em bytes, reservada para o uso de não dados. Por padrão, o nó do Redis OSS crescerá até consumir o <code>maxmemory</code> do nó (consulte Parâmetros específicos de node-type do Redis OSS). Se isso ocorrer, o desempenho do nó provavelmente sofrerá devido a uma paginação de memória excessiva. Ao reservar memória, você pode separar uma parte da memória disponível para fins não Redis OSS, a fim de ajudar a reduzir a quantidade de paginação. Esse parâmetro é específico e não faz parte da distribuição padrão do Redis OSS. ElastiCache Para obter mais informações, consulte <code>reserved-memory-percent</code> e Gerenciamento de memória reservada para Valkey e Redis OSS.
<code>set-max-intset-entries</code>	Padrão: 512 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória utilizada para certos tipos de conjuntos (strings que são inteiros em radix 10 no intervalo de inteiros de 64 bits com sinal). Esses conjuntos com menos que o número especificado de entradas são armazenados usando uma codificação especial que economiza espaço.

Nome	Detalhes	Description
slave-all ow-chaini ng	Padrão: não Tipo: string Modificável: não	Determina se uma réplica de leitura no Redis OSS pode ter réplicas de leitura próprias.
slowlog-l og-slower -than	Padrão: 10000 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O tempo máximo de execução, em microssegundos, para que os comandos sejam registrados pelo atributo de log lento do Redis OSS.
slowlog-m ax-len	Padrão: 128 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O comprimento máximo do log lento do Redis OSS.
tcp-keepa live	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Se estiver definido como um valor diferente de zero (N), os clientes do nó são sondados a cada N segundos para garantir que ainda estejam conectados. Com a configuração padrão de 0, essa sondagem não ocorre.  Important Alguns aspectos desse parâmetro mudaram na versão 3.2.4 do Redis OSS. Consulte Parâmetros alterados no Redis OSS 3.2.4 (aprimorado).

Nome	Detalhes	Description
<code>timeout</code>	Padrão: 0 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O número de segundos que um nó espera antes do tempo limite. Os valores são: • 0 – nunca desconectar um cliente ocioso. • 1-19 – valores inválidos. • ≥ 20 – o número de segundos que um nó espera antes de desconectar um cliente ocioso.
<code>zset-max-ziplist-entries</code>	Padrão: 128 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória utilizada para conjuntos classificados. Os conjuntos classificados com menos que o número especificado de elementos são armazenados usando uma codificação especial que economiza espaço.
<code>zset-max-ziplist-value</code>	Padrão: 64 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	Determina a quantidade de memória utilizada para conjuntos classificados. Os conjuntos classificados com entradas menores que o número especificado de bytes são armazenados usando uma codificação especial que economiza espaço.

Note

Se você não especificar um grupo de parâmetros para seu cluster do Redis OSS 2.6.13, um grupo de parâmetros padrão (`default.redis2.6`) será usado. Não é possível alterar os valores de parâmetros no grupo de parâmetros padrão. No entanto, você sempre pode criar um grupo de parâmetros personalizado e atribuí-lo ao seu cluster a qualquer momento.

Parâmetros específicos de node-type do Redis OSS

Embora a maioria dos parâmetros tenha um valor único, alguns parâmetros têm valores diferentes dependendo do tipo de nó usado. A tabela a seguir mostra os valores padrão para os parâmetros `maxmemory`, `client-output-buffer-limit-slave-hard-limit` e `client-output-buffer-limit-slave-soft-limit` para cada tipo de nó. O valor de `maxmemory` é o número máximo de bytes disponíveis para uso, dados e outros usos no nó. Para obter mais informações, consulte [Available memory](#) (Memória disponível).

Note

O parâmetro `maxmemory` não pode ser modificado.

Tipo de nó	Maxmemory	Client-output-buffer-limit - slave-hard-limit	Client-output-buffer-limit - slave-soft-limit
cache.t1.micro	142606336	14260633	14260633
cache.t2.micro	581959680	58195968	58195968
cache.t2.small	1665138688	166513868	166513868
cache.t2.medium	3461349376	346134937	346134937
cache.t3.micro	536870912	53687091	53687091
cache.t3.small	1471026299	147102629	147102629
cache.t3.medium	3317862236	331786223	331786223
cache.t4g.micro	536870912	53687091	53687091
cache.t4g.small	1471026299	147102629	147102629
cache.t4g.medium	3317862236	331786223	331786223
cache.m1.small	943718400	94371840	94371840

Tipo de nó	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.m1.medium	3093299200	309329920	309329920
cache.m1.large	7025459200	702545920	702545920
cache.m1.xlarge	14889779200	1488977920	1488977920
cache.m2.xlarge	17091788800	1709178880	1709178880
cache.m2.2xlarge	35022438400	3502243840	3502243840
cache.m2.4xlarge	70883737600	7088373760	7088373760
cache.m3.medium	2988441600	309329920	309329920
cache.m3.large	6501171200	650117120	650117120
cache.m3.xlarge	14260633600	1426063360	1426063360
cache.m3.2xlarge	29989273600	2998927360	2998927360
cache.m4.large	6892593152	689259315	689259315
cache.m4.xlarge	15328501760	1532850176	1532850176
cache.m4.2xlarge	31889126359	3188912636	3188912636
cache.m4.4xlarge	65257290629	6525729063	6525729063
cache.m4.10xlarge	166047614239	16604761424	16604761424
cache.m5.large	6854542746	685454275	685454275
cache.m5.xlarge	13891921715	1389192172	1389192172
cache.m5.2xlarge	27966669210	2796666921	2796666921
cache.m5.4xlarge	56116178125	5611617812	5611617812

Tipo de nó	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.m5.12xlarge	168715971994	16871597199	16871597199
cache.m5.24xlarge	337500562842	33750056284	33750056284
cache.m6g.large	6854542746	685454275	685454275
cache.m6g.xlarge	13891921715	1389192172	1389192172
cache.m6g.2xlarge	27966669210	2796666921	2796666921
cache.m6g.4xlarge	56116178125	5611617812	5611617812
cache.m6g.8xlarge	111325552312	11132555231	11132555231
cache.m6g.12xlarge	168715971994	16871597199	16871597199
cache.m6g.16xlarge	225000375228	22500037523	22500037523
cache.c1.xlarge	6501171200	650117120	650117120
cache.r3.large	14470348800	1468006400	1468006400
cache.r3.xlarge	30513561600	3040870400	3040870400
cache.r3.2xlarge	62495129600	6081740800	6081740800
cache.r3.4xlarge	126458265600	12268339200	12268339200
cache.r3.8xlarge	254384537600	24536678400	24536678400
cache.r4.large	13201781556	1320178155	1320178155
cache.r4.xlarge	26898228839	2689822883	2689822883
cache.r4.2xlarge	54197537997	5419753799	5419753799
cache.r4.4xlarge	108858546586	10885854658	10885854658

Tipo de nó	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.r4.8xlarge	218255432090	21825543209	21825543209
cache.r4.16xlarge	437021573120	43702157312	43702157312
cache.r5.large	14037181030	1403718103	1403718103
cache.r5.xlarge	28261849702	2826184970	2826184970
cache.r5.2xlarge	56711183565	5671118356	5671118356
cache.r5.4xlarge	113609865216	11360986522	11360986522
cache.r5.12xlarge	341206346547	34120634655	34120634655
cache.r5.24xlarge	682485973811	68248597381	68248597381
cache.r6g.large	14037181030	1403718103	1403718103
cache.r6g.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r6gd.xlarge	28261849702	2826184970	2826184970
cache.r6gd.2xlarge	56711183565	5671118356	5671118356
cache.r6gd.4xlarge	113609865216	11360986522	11360986522
cache.r6gd.8xlarge	225000375228	22500037523	22500037523

Tipo de nó	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.r6gd.12xlarge	341206346547	34120634655	34120634655
cache.r6gd.16xlarge	450000750456	45000075046	45000075046
cache.r7g.large	14037181030	1403718103	1403718103
cache.r7g.xlarge	28261849702	2826184970	2826184970
cache.r7g.2xlarge	56711183565	5671118356	5671118356
cache.r7g.4xlarge	113609865216	11360986522	11360986522
cache.r7g.8xlarge	225000375228	22500037523	22500037523
cache.r7g.12xlarge	341206346547	34120634655	34120634655
cache.r7g.16xlarge	450000750456	45000075046	45000075046
cache.m7g.large	6854542746	685454275	685454275
cache.m7g.xlarge	13891921715	1389192172	1389192172
cache.m7g.2xlarge	27966669210	2796666921	2796666921
cache.m7g.4xlarge	56116178125	5611617812	5611617812
cache.m7g.8xlarge	111325552312	11132555231	11132555231
cache.m7g.12xlarge	168715971994	16871597199	16871597199
cache.m7g.16xlarge	225000375228	22500037523	22500037523
cache.c7gn.large	3317862236	1403718103	1403718103
cache.c7gn.xlarge	6854542746	2826184970	2826184970
cache.c7gn.2xlarge	13891921715	5671118356	5671118356

Tipo de nó	Maxmemory	C lient-output-buffe r-limit - slave-hard- limit	C lient-output-buffe r-limit - slave-soft- limit
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

Note

Todos os tipos de instâncias da geração atual são criados em uma Amazon Virtual Private Cloud VPC por padrão.

As instâncias T1 não oferecem suporte a Multi-AZ.

As instâncias T1 e T2 não oferecem suporte ao AOF do Redis OSS.

As variáveis de configuração do Redis OSS `appendonly` e `appendfsync` não são compatíveis com o Redis OSS versão 2.8.22 e posteriores.

Parâmetros específicos do Memcached

Memcached

Se você não especificar um parameter group para seu cluster Memcached, será usado um parameter group padrão apropriado. Não é possível alterar os valores de nenhum parâmetro em um grupo de parâmetros padrão. No entanto, é possível criar um grupo de parâmetros personalizado e atribuí-lo ao cluster a qualquer momento. Para obter mais informações, consulte [Criação de um grupo de parâmetros do ElastiCache](#).

Tópicos

- [Mudanças no Memcached 1.6.17](#)
- [Parâmetros adicionados do Memcached 1.6.6](#)
- [Alterações de parâmetros do Memcached 1.5.10](#)
- [Parâmetros adicionados do Memcached 1.4.34](#)

- [Parâmetros adicionados do Memcached 1.4.33](#)
- [Parâmetros adicionados do Memcached 1.4.24](#)
- [Parâmetros adicionados do Memcached 1.4.14](#)
- [Parâmetros com suporte do Memcached 1.4.5](#)
- [Sobrecarga de conexões do Memcached](#)
- [Parâmetros específicos do tipo de nó do Memcached](#)

Mudanças no Memcached 1.6.17

A partir do Memcached 1.6.17, não oferecemos mais suporte a estes comandos administrativos: `lru_crawler`, `lru` e `slabs`. Com essas mudanças, você não poderá fazer `enable/disable lru_crawler` em tempo de execução por meio de comandos. `enable/disable lru_crawler` Por favor, modificando seu grupo de parâmetros personalizado.

Parâmetros adicionados do Memcached 1.6.6

Para o Memcached 1.6.6, não há suporte para parâmetros adicionais.

Família de grupos de parâmetros: `memcached1.6`

Alterações de parâmetros do Memcached 1.5.10

Para o Memcached 1.5.10, há suporte para os seguintes parâmetros adicionais.

Família de parameter groups: `memcached1.5`

Nome	Detalhes	Description
no_modern	Padrão: 1	Um alias para desabilitar comandos <code>slab_reassign</code> , <code>lru_maintainer_thread</code> , <code>lru_segmented</code> e <code>maxconns_fast</code> .
	Tipo: booleano	
	Modificável: sim	
	Valores permitidos: 0,1	
	As alterações terão efeito: na execução	
		Ao usar o Memcached 1.5 e posteriores, <code>no_modern</code> também

Nome	Detalhes	Description
		define o <code>hash_algorithm</code> como <code>jenkins</code>. Além disso, ao usar o Memcached 1.5.10, <code>inline_ascii_reponse</code> é controlado pelo parâmetro <code>parallelly</code>. Isso significa que, se <code>no_modern</code> estiver desabilitado, <code>inline_ascii_reponse</code> estará desabilitado. Do mecanismo Memcached 1.5.16 em diante, o parâmetro <code>inline_ascii_response</code> não se aplica mais, portanto, <code>no_modern</code> estar habilitado ou desabilitado não tem efeito sobre <code>inline_ascii_response</code>. Se <code>no_modern</code> estiver desabilitado, então <code>slab_reassign</code>, <code>lru_maintainer_thread</code>, <code>lru_segmented</code> e <code>maxconns_fast</code> ESTARÃO habilitados. Como os parâmetros <code>slab_auto_move</code> e <code>hash_algorithm</code> não são parâmetros SWITCH, sua configuração é baseada nas configurações do grupo de parâmetros. Se você quiser desabilitar <code>no_modern</code> e reverter para <code>modern</code>, é preciso configurar um grupo de parâmetros personali

Nome	Detalhes	Description
		zado para desabilitar esse parâmetro e, em seguida, reinicializar para que essas alterações entrem em vigor.  Note O valor padrão de configuração para esse parâmetro foi alterado de 0 para 1 a partir de 20 de agosto de 2021. O valor padrão atualizado será automaticamente coletado por novos ElastiCache usuários em cada região após 20 de agosto de 2021. ElastiCache Os usuários existentes nas regiões antes de 20 de agosto de 2021 precisam modificar manualmente seus grupos de parâmetros personalizados para aceitar essa nova alteração.

Nome	Detalhes	Description
<code>inline_ascii_resp</code>	Padrão: 0 Tipo: booliano Modificável: sim Valores permitidos: 0,1 As alterações terão efeito: na execução	Armazena números da resposta VALUE, dentro de um item, usando até 24 bytes. Pequena desaceleração para os conjuntos do ASCII get, faster.

Para a versão 1.5.10 do Memcached, os parâmetros seguintes foram removidos.

Nome	Detalhes	Description
<code>expirezero_does_no_t_evict</code>	Padrão: 0 Tipo: booliano Modificável: sim Valores permitidos: 0,1 As alterações terão efeito: na execução	Não é mais compatível nesta versão.
<code>modern</code>	Padrão: 1 Tipo: booliano Modificável: sim (requer reinicialização se definido para <code>no_modern</code>)	Não é mais compatível nesta versão. A partir desta versão, o <code>no-modern</code> é ativado por padrão com cada inicialização ou reinicialização.

Nome	Detalhes	Description
	Valores permitidos: 0,1 As alterações terão efeito: na execução	

Parâmetros adicionados do Memcached 1.4.34

Para o Memcached 1.4.34, não há suporte para parâmetros adicionais.

Família de parameter groups: memcached1.4

Parâmetros adicionados do Memcached 1.4.33

Para o Memcached 1.4.33, há suporte para os seguintes parâmetros adicionais.

Família de parameter groups: memcached1.4

Nome	Detalhes	Description
modern	Padrão: habilitado Tipo: booliano Modificável: sim As alterações terão efeito: na execução	Um alias para vários recursos. Habilitar modern equivale a ativar os seguintes comandos e usar um algoritmo de hash murmur3: <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> e <code>hash_algorithm=murmur3</code> .
watch	Padrão: habilitado Tipo: booliano Modificável: sim	Buscas, remoções ou mutações de logs. Quando, por exemplo, o usuário ativa watch, ele pode ver logs quando ocorre <code>get</code> , <code>set</code> , <code>delete</code> ou <code>update</code> .

Nome	Detalhes	Description
	As alterações terão efeito: imediatamente Logs podem ser descartados se o usuário atingir seus limites de <code>watcher_logbuf_size</code> e <code>worker_logbuf_size</code>.	
<code>idle_timeout</code>	Padrão: 0 (desabilitado) Tipo: inteiro Modificável: sim As alterações terão efeito: na execução	O número mínimo de segundos que um cliente terá permissão para ficar ocioso antes que o seu fechamento seja solicitado. Faixa de valores: 0 a 86400.
<code>track_sizes</code>	Padrão: desabilitado Tipo: booliano Modificável: sim As alterações terão efeito: na execução	Mostra os tamanhos que cada grupo de slabs consumiu. Habilitar <code>track_sizes</code> permite que você execute <code>stats sizes</code> sem a necessidade de executar <code>stats sizes enable</code>.

Nome	Detalhes	Description
<code>watcher_logbuf_size</code>	Padrão: 256 (KB) Tipo: inteiro Modificável: sim As alterações terão efeito: na execução	O comando <code>watch</code> ativa o registro em log de streams para o Memcached. No entanto, <code>watch</code> poderá descartar logs se a taxa de remoções, mutações ou buscas for alta o suficiente para fazer com que o buffer de registro em log fique cheio. Nessas situações, os usuários podem aumentar o tamanho do buffer para reduzir a chance de perdas de log.
<code>worker_logbuf_size</code>	Padrão: 64 (KB) Tipo: inteiro Modificável: sim As alterações terão efeito: na execução	O comando <code>watch</code> ativa o registro em log de streams para o Memcached. No entanto, <code>watch</code> poderá descartar logs se a taxa de remoções, mutações ou buscas for alta o suficiente para fazer com que o buffer de registro em log fique cheio. Nessas situações, os usuários podem aumentar o tamanho do buffer para reduzir a chance de perdas de log.
<code>slab_chunk_max</code>	Padrão: 524288 (bytes) Tipo: inteiro Modificável: sim As alterações terão efeito: na execução	Especifica o tamanho máximo de um slab. A configuração de tamanho de slab menor usa a memória de forma mais eficiente. Itens com mais de <code>slab_chunk_max</code> são divididos em vários slabs.

Nome	Detalhes	Description
<code>lru_crawler metadump</code> <code>[all 1 2 3]</code>	Padrão: desabilitado Tipo: booliano Modificável: sim As alterações terão efeito: imediatamente	se <code>lru_crawler</code> estiver habilitado, esse comando despejará todas as chaves. <code>all 1 2 3</code> - todos os slab, ou especifique um número específico de slab

Parâmetros adicionados do Memcached 1.4.24

Para o Memcached 1.4.24, há suporte para os seguintes parâmetros adicionais.

Família de parameter groups: `memcached1.4`

Nome	Detalhes	Description
<code>disable_flush_all</code>	Padrão: 0 (desabilitado) Tipo: booliano Modificável: sim As alterações terão efeito: na execução	Adicione o parâmetro (-F) para desabilitar <code>flush_all</code> . Útil se você nunca quiser ser capaz de executar uma liberação total em instâncias de produção. Valores: 0, 1 (o usuário pode fazer <code>flush_all</code> quando o valor é 0).
<code>hash_algorithm</code>	Padrão: jenkins Tipo: string Modificável: sim As alterações terão efeito: na execução	O algoritmo de hash a ser usado. Valores permitidos: <code>murmur3</code> e <code>jenkins</code> .
<code>lru_crawler</code>	Padrão: 0 (desabilitado)	

Nome	Detalhes	Description
	Tipo: booliano Modificável: sim As alterações terão efeito: após a reinicialização  Note Você pode habilitar <code>lru_crawler</code> temporariamente em tempo de execução a partir da linha de comando. Para obter mais informações, consulte a coluna Descrição.	Limpa as classes de itens de slabs que expiraram. Este é um processo de baixo impacto que é executado em segundo plano. Atualmente, ele exige iniciar um rastreamento usando um comando manual. Para habilitar temporariamente, execute <code>lru_crawler enable</code> na linha de comando. <code>lru_crawler 1,3,5</code> rastreia as classes de slab 1, 3 e 5, procurando itens expirados para adicionar à lista livre. Valores: 0,1  Note Habilitar <code>lru_crawler</code> na linha de comando habilita o crawler até que ele seja desabilitado na linha de comando ou na próxima reinicialização. Para habilitar permanentemente, você deve modificar o valor do parâmetro. Para obter mais informações, consulte Modificação de um grupo de parâmetros do ElastiCache.

Nome	Detalhes	Description
<code>lru_maintainer</code>	Padrão: 0 (desabilitado) Tipo: booleano Modificável: sim As alterações terão efeito: na execução	Uma linha de fundo que embaralha os itens entre eles à LRUs medida que as capacidades são atingidas. Valores: 0, 1.
<code>expirezero_does_not_evict</code>	Padrão: 0 (desabilitado) Tipo: booleano Modificável: sim As alterações terão efeito: na execução	Quando usado com <code>lru_maintainer</code>, faz com que itens com um tempo de expiração de 0 não possam ser removidos.  Warning Isso pode lotar a memória disponível para outros itens removíveis. Pode ser configurado para desconsiderar <code>lru_maintainer</code>.

Parâmetros adicionados do Memcached 1.4.14

Para o Memcached 1.4.14, há suporte para os seguintes parâmetros adicionais.

Família de parameter groups: memcached1.4

Parâmetros adicionados no Memcached 1.4.14

Nome	Description
config_max	O número máximo de entradas de ElastiCache configuração.
config_size_max	O tamanho máximo das entradas de configuração, em bytes.
hashpower_init	O tamanho inicial da tabela de ElastiCache hash, expresso como uma potência de dois. O padrão é 16 (2^{16}) ou 65536 chaves.
maxconns_fast	Altera o modo como novas solicitações de conexão são tratadas quando o limite máximo de conexão é atingido. Se esse parâmetro estiver definido como 0 (zero), novas conexões serão adicionadas à fila de backlog e aguardarão até que outras conexões sejam encerradas. Se o parâmetro for definido como 1, ElastiCache envia um erro ao cliente e fecha imediatamente a conexão.

Nome	Description
<code>slab_automove</code>	Ajusta o algoritmo de movimentação automática de slabs: se esse parâmetro estiver definido como 0 (zero), o algoritmo de movimentação automática será desabilitado. Se estiver definido como 1, o ElastiCache usará uma abordagem lenta e conservadora para mover slabs automaticamente. Se estiver definido como 2, move ElastiCache agressivamente as placas sempre que houver um despejo. (Esse modo não é recomendado, exceto para fins de teste.)

Nome	Description
slab_reassign	Habilite ou desabilite a reatribuição de slabs. Se esse parâmetro estiver definido como 1, você poderá usar o comando "slabs reassign" para reatribuir a memória manualmente.

Parâmetros com suporte do Memcached 1.4.5

Família de parameter groups: memcached1.4

Para o Memcached 1.4.5, há suporte para os seguintes parâmetros.

Parâmetros adicionados no Memcached 1.4.5

Nome	Detalhes	Description
backlog_queue_limit	Padrão: 1024 Tipo: inteiro Modificável: não	O limite de fila de backlog.

Nome	Detalhes	Description
<code>binding_protocol</code>	Padrão: automático Tipo: string Modificável: sim As alterações terão efeito: após a reinicialização	O protocolo de associação. Os valores permitidos são: <code>ascii</code> e <code>auto</code>. Para obter orientação sobre como modificar o valor de <code>binding_protocol</code>, consulte Modificação de um grupo de parâmetros do ElastiCache.
<code>cas_disabled</code>	Padrão: 0 (<code>false</code>) Tipo: booleano Modificável: sim As alterações terão efeito: após a reinicialização	Se 1 (<code>true</code>), as operações Verificar e Definir (CAS) serão desabilitadas, e os itens armazenados consumirão 8 bytes a menos do que quando essas operações estão habilitadas.
<code>chunk_size</code>	Padrão: 48 Tipo: inteiro Modificável: sim As alterações terão efeito: após a reinicialização	O valor mínimo, em bytes, de espaço para alocar para a chave, o valor e os sinalizadores do menor item.
<code>chunk_size_growth_factor</code>	Padrão: 1.25 Tipo: flutuante Modificável: sim As alterações terão efeito: após a reinicialização	O fator de crescimento que controla o tamanho de cada bloco Memcached sucessivo; cada bloco será <code>chunk_size_growth_factor</code> vezes maior que o bloco anterior.

Nome	Detalhes	Description
<code>error_on_memory_exhausted</code>	Padrão: 0 (false) Tipo: booliano Modificável: sim As alterações terão efeito: após a reinicialização	Se 1 (true), quando não há mais memória para armazenar itens, o Memcached retornará um erro em vez de remover itens.
<code>large_memory_pages</code>	Padrão: 0 (false) Tipo: booliano Modificável: não	Se 1 (verdadeiro), ElastiCache tentará usar páginas de memória grandes.
<code>lock_down_paged_memory</code>	Padrão: 0 (false) Tipo: booliano Modificável: não	Se 1 (verdadeiro), ElastiCache bloqueará toda a memória paginada.
<code>max_item_size</code>	Padrão: 1048576 Tipo: inteiro Modificável: sim As alterações terão efeito: após a reinicialização	O tamanho, em bytes, do maior item que pode ser armazenado no cluster.
<code>max_simultaneous_connections</code>	Padrão: 65000 Tipo: inteiro Modificável: não	O número máximo de conexões simultâneas.

Nome	Detalhes	Description
<code>maximize_core_file_limit</code>	Padrão: 0 (false) Tipo: booliano Permite modificação: As alterações terão efeito: após a reinicialização	Se 1 (verdadeiro), ElastiCache maximizará o limite do arquivo principal.
<code>memcached_connections_overhead</code>	Padrão: 100 Tipo: inteiro Modificável: sim As alterações terão efeito: após a reinicialização	A quantidade de memória a ser reservada para conexões Memcached e outras sobrecargas diversas. Para obter informações sobre esse parâmetro, consulte Sobrecarga de conexões do Memcached .
<code>requests_per_event</code>	Padrão: 20 Tipo: inteiro Modificável: não	O número máximo de solicitações por evento para uma determinada conexão. Este limite é necessário para evitar o esgotamento de recursos.

Sobrecarga de conexões do Memcached

Em cada nó, a memória disponibilizada para armazenar itens é a memória total disponível nesse nó (que é armazenada no parâmetro `max_cache_memory`) menos a memória usada para conexões e outras sobrecargas (que é armazenada no parâmetro `memcached_connections_overhead`). Por exemplo, um nó do tipo `cache.m1.small` tem um `max_cache_memory` de 1300 MB. Com o valor padrão `memcached_connections_overhead` de 100 MB, o processo Memcached terá 1200 MB disponíveis para armazenar itens.

Os valores padrão para o parâmetro `memcached_connections_overhead` satisfazem a maioria dos casos de uso. No entanto, a quantidade necessária de alocação para sobrecarga de conexão pode variar dependendo de vários fatores, entre eles a taxa de solicitação, o tamanho da carga útil e o número de conexões.

Você pode alterar o valor de `memcached_connections_overhead` para melhor atender às necessidades do seu aplicativo. Por exemplo, aumentar o valor do parâmetro `memcached_connections_overhead` reduzirá a quantidade de memória disponível para armazenar itens e fornecerá um buffer maior para a sobrecarga de conexões. Diminuir o valor do parâmetro `memcached_connections_overhead` fornecerá mais memória para armazenar itens, mas pode aumentar seu risco de uso de permuta e desempenho degradado. Se você observar uso de permuta e desempenho degradado, tente aumentar o valor do parâmetro `memcached_connections_overhead`.

 Important

Para o tipo de nó `cache.t1.micro`, o valor para `memcached_connections_overhead` é determinado da seguinte forma:

- Se o cluster estiver usando o grupo de parâmetros padrão, ElastiCache definirá o valor `memcached_connections_overhead` para 13 MB.
- Se o seu cluster estiver usando um `parameter group` que você criou, será possível definir o valor de `memcached_connections_overhead` como um valor de sua escolha.

Parâmetros específicos do tipo de nó do Memcached

Embora a maioria dos parâmetros tenha um valor único, alguns parâmetros têm valores diferentes dependendo do tipo de nó usado. A tabela a seguir mostra os valores padrão para os parâmetros `max_cache_memory` e `num_threads` para cada tipo de nó. Os valores nesses parâmetros não podem ser modificados.

Tipo de nó	<code>max_cache_memory</code> (em megabytes)	<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
<code>cache.t3.micro</code>	512	2

Tipo de nó	max_cache_memory (em megabytes)	num_threads
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	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
cache.m4.4xlarge	62234	16

Tipo de nó	max_cache_memory (em megabytes)	num_threads
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
cache.r3.8xlarge	120600	32

Tipo de nó	max_cache_memory (em megabytes)	num_threads
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
cache.c7gn.large	3164	2

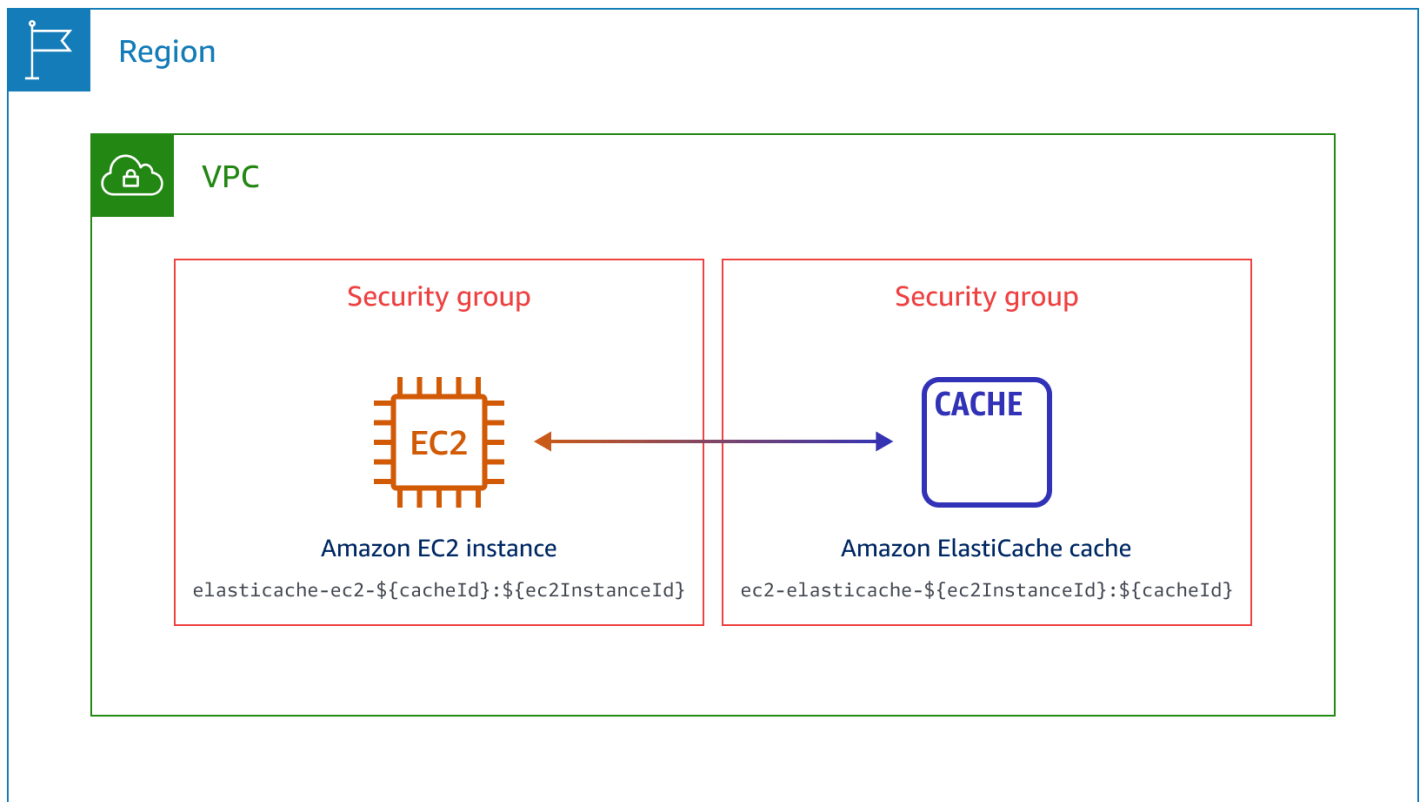
Tipo de nó	max_cache_memory (em megabytes)	num_threads
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

 Note

Todas as instâncias T2 são criadas em uma Amazon Virtual Private Cloud (Amazon VPC).

Conectar uma instância do EC2 e um cache do ElastiCache automaticamente

Você pode usar o console do ElastiCache para simplificar a configuração de uma conexão entre uma instância do Amazon Elastic Compute Cloud (Amazon EC2) e um cache do ElastiCache. Geralmente, seu cache está em uma sub-rede privada e sua instância do EC2 está em uma sub-rede pública em uma VPC. Você pode utilizar um cliente SQL em sua instância do EC2 para se conectar ao seu cache do ElastiCache. A instância do EC2 também pode executar servidores web ou aplicações que acessam seu cache privado do ElastiCache.



Tópicos

- [Conectividade automática com uma instância do EC2](#)
- [Visualizar recursos computacionais conectados](#)

Conectividade automática com uma instância do EC2

Quando você configura uma conexão entre uma instância do EC2 e um cache do ElastiCache, o ElastiCache automaticamente configura o grupo de segurança da VPC para sua instância do EC2 e para seu cache do ElastiCache.

Confira a seguir os requisitos para conectar uma instância do EC2 ao cache do ElastiCache:

- A instância do EC2 deve existir na mesma VPC do cache do ElastiCache.

Se não houver nenhuma instância do EC2 na mesma VPC, o console fornecerá um link para que você crie uma.

- O usuário que configura a conectividade deve ter permissões para realizar as seguintes operações do Amazon EC2. Essas permissões geralmente são adicionadas às contas do EC2 quando elas

são criadas. Para obter mais informações sobre permissões do EC2, consulte [Como conceder as permissões necessárias para os recursos do Amazon EC2](#).

- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Quando você configura uma conexão com uma instância do EC2, o ElastiCache atua de acordo com a configuração atual dos grupos de segurança associados ao cache do ElastiCache e à instância do EC2, conforme descrito na tabela a seguir.

Configuração atual do grupo de segurança do ElastiCache	Configuração atual do grupo de segurança do EC2	Ação do ElastiCache
Há um ou mais grupos de segurança associados ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do cache do ElastiCache como origem.	O ElastiCache não realiza nenhuma ação. Uma conexão já foi configurada automaticamente entre a instância do EC2 e o cache do ElastiCache. Como já existe uma conexão entre a instância do EC2 e o cache do ElastiCache, os grupos de segurança não são modificados.
Qualquer uma das seguintes condições se aplica:	Qualquer uma das seguintes condições se aplica:	ELC action: create new security groups

Configuração atual do grupo de segurança do ElastiCache	Configuração atual do grupo de segurança do EC2	Ação do ElastiCache
- Não há um grupo de segurança associado ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code>. - Há um ou mais grupos de segurança associados ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code>. No entanto, o ElastiCache não pode usar nenhum desses grupos de segurança para a conexão com a instância do EC2. O ElastiCache não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da instância do EC2 como origem. O ElastiCache também não pode usar um grupo de segurança que tenha sido modificado. São exemplos de modificação a adição de uma regra ou a alteração da porta de uma regra existente.	- Não há um grupo de segurança associado à instância do EC2 com um nome que corresponde ao padrão <code>ec2-elasticache-\${ec2InstanceId}:\${cacheId}</code>. - Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão <code>ec2-elasticache-\${ec2InstanceId}:\${cacheId}</code>. No entanto, o ElastiCache não pode usar nenhum desses grupos de segurança para a conexão com o cache do ElastiCache. O ElastiCache não pode usar um grupo de segurança que não tenha uma regra de saída no grupo de segurança da VPC do cache do ElastiCache como origem. O ElastiCache também não pode usar um grupo de segurança que tenha sido modificado.	

Configuração atual do grupo de segurança do ElastiCache	Configuração atual do grupo de segurança do EC2	Ação do ElastiCache
Há um ou mais grupos de segurança associados ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . No entanto, o ElastiCache não pode usar nenhum desses grupos de segurança para a conexão com o cache do ElastiCache. O ElastiCache não pode usar um grupo de segurança que não tenha uma regra de saída no grupo de segurança da VPC do cache do ElastiCache como origem. O ElastiCache também não pode usar um grupo de segurança que tenha sido modificado.	ELC action: create new security groups

Configuração atual do grupo de segurança do ElastiCache	Configuração atual do grupo de segurança do EC2	Ação do ElastiCache
Há um ou mais grupos de segurança associados ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code> . Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC da instância do EC2 como origem.	Existe um grupo de segurança do EC2 válido para a conexão, mas ele não está associado à instância do EC2. Esse grupo de segurança tem um nome que corresponde ao padrão <code>ec2-elasticache-\${ec2InstanceId}:\${cacheId}</code> . Não foi modificado. Ele tem apenas uma regra de saída com o grupo de segurança da VPC do cache do ElastiCache como origem.	ELC action: associate EC2 security group

Configuração atual do grupo de segurança do ElastiCache	Configuração atual do grupo de segurança do EC2	Ação do ElastiCache
Qualquer uma das seguintes condições se aplica: - Não há um grupo de segurança associado ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code>. - Há um ou mais grupos de segurança associados ao cache do ElastiCache com um nome correspondente ao padrão <code>elasticache-ec2-\${cacheId}:\${ec2InstanceId}</code>. No entanto, o ElastiCache não pode usar nenhum desses grupos de segurança para a conexão com a instância do EC2. O ElastiCache não pode usar um grupo de segurança que não tenha uma regra de entrada no grupo de segurança da VPC da instância do EC2 como origem. O ElastiCache também não pode usar um grupo de segurança que tenha sido modificado.	Há um ou mais grupos de segurança associados à instância do EC2 com um nome que corresponde ao padrão <code>ec2-elasticache-\${ec2InstanceId}:\${cacheId}</code>. Um grupo de segurança que corresponde ao padrão não foi modificado. Esse grupo de segurança tem apenas uma regra de saída com o grupo de segurança da VPC do cache do ElastiCache como origem.	ELC action: create new security groups

Ação do ElastiCache: criar novos grupos de segurança

O ElastiCache realiza as seguintes ações:

- Cria um grupo de segurança que corresponde ao padrão `elasticache-ec2-${cacheId}:${ec2InstanceId}`. Esse grupo de segurança tem uma regra de entrada com o grupo de segurança da VPC da instância do EC2 como origem. Esse grupo de segurança está associado ao cache do ElastiCache e permite que a instância do EC2 o acesse.
- Cria um grupo de segurança que corresponde ao padrão `elasticache-ec2-${cacheId}:${ec2InstanceId}`. Esse grupo de segurança tem uma regra de saída com o grupo de segurança da VPC do cache do ElastiCache como destino. Esse grupo de segurança está associado à instância do EC2 e permite que ela envie tráfego ao cache do ElastiCache.

Ação do ElastiCache: associar o grupo de segurança do EC2

O ElastiCache associa o grupo de segurança do EC2 existente e válido à instância do EC2. Esse grupo de segurança permite que a instância do EC2 envie tráfego ao cache do ElastiCache.

Visualizar recursos computacionais conectados

Você pode usar o Console de gerenciamento da AWS para visualizar os recursos computacionais conectados a um cache do ElastiCache. Os recursos mostrados incluem conexões de recursos computacionais que foram configuradas automaticamente. Por exemplo, você pode permitir que um recurso computacional acesse um cache manualmente adicionando uma regra ao grupo de segurança da VPC associado ao cache. Esses recursos não aparecerão na lista de recursos computacionais conectados.

Para que um recurso computacional seja listado, as mesmas condições devem ser atendidas quando conectar automaticamente uma instância do EC2 e um cache do ElastiCache.

Para visualizar recursos computacionais conectados a um cache do ElastiCache

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache
2. No painel de navegação, escolha Cache e, em seguida, escolha um cache do Valkey ou Redis OSS.
3. Na guia Connectivity & security (Conectividade e segurança), veja os recursos computacionais em Set up compute connection (Configurar conexão de computação).

Connected compute resources (1)
Set up compute connection

Connections to compute resources that were automatically created by ElastiCache for your cache are shown here. Connections to compute resources that were manually created are not displayed.

< 1 > ⚙

Compute resource name	Status	Resource type	Availability zones	Security group
i-0bdb4ea9f9bb36aec	Running	t2.micro	eu-west-3c	ec2-elasticache-i-0bdb4ea9f9bb36aec:cache-foo

Dimensionamento ElastiCache

Você pode escalar seu ElastiCache cache para atender às suas necessidades. Caches com tecnologia sem servidor e clusters baseados em nós oferecem várias opções de escalabilidade diferentes.

Dimensionamento sem servidor ElastiCache

ElastiCache O Serverless acomoda automaticamente seu tráfego de carga de trabalho à medida que ele aumenta ou diminui. Para cada cache ElastiCache sem servidor, rastreia ElastiCache continuamente a utilização de recursos como CPU, memória e rede. Quando qualquer um desses recursos está restrito, o ElastiCache Serverless se expande adicionando um novo fragmento e redistribuindo os dados ao novo fragmento, sem nenhum tempo de inatividade para seu aplicativo. Você pode monitorar os recursos consumidos pelo seu cache CloudWatch monitorando a BytesUsedForCache métrica para armazenamento de dados em cache e ElastiCacheProcessingUnits (ECPU) para uso de computação.

Definição dos limites de escalabilidade para gerenciar custos

Você pode optar por configurar um uso máximo no armazenamento de dados em cache e no seu cache ECPU/second para controlar os custos do cache. Isso vai garantir que o uso do cache jamais exceda o máximo configurado.

Se você definir uma escala máxima, a aplicação pode apresentar um desempenho de cache reduzido quando o cache atinge o máximo. Quando você define um máximo de armazenamento de dados em cache e seu armazenamento de dados em cache atinge o máximo, ElastiCache começará a despejar dados em seu cache que tenham um Time-To-Live (TTL) definido, usando a lógica LRU. Se não houver dados que possam ser removidos, as solicitações de gravação de dados adicionais receberão uma mensagem de erro de falta de memória (OOM). Quando você define um ECPU/second máximo e a utilização computacional de sua carga de trabalho excede esse valor, ElastiCache começará a limitar as solicitações.

Se você configurar um limite máximo em `BytesUsedForCache` ou `ElastiCacheProcessingUnits`, é altamente recomendável configurar um CloudWatch alarme com um valor inferior ao limite máximo para que você seja notificado quando seu cache estiver operando perto desses limites. É recomendável definir um alarme em 75% do limite máximo definido. Consulte a documentação sobre como configurar CloudWatch alarmes.

Pré-escalonamento com Serverless ElastiCache

ElastiCache Pré-escalonamento sem servidor

Com o pré-escalonamento, também chamado de pré-aquecimento, você pode definir limites mínimos suportados para seu cache. ElastiCache Você pode definir esses mínimos para unidades ElastiCache de processamento (ECPUs) por segundo ou armazenamento de dados. Isso pode ser útil na preparação para eventos de escalonamento previstos. Por exemplo, se uma empresa de jogos espera um aumento de 5 vezes nos logins no primeiro minuto de lançamento do novo jogo, ela pode preparar seu cache para esse aumento significativo no uso.

Você pode realizar o pré-escalonamento usando o ElastiCache console, a CLI ou a API. ElastiCache O Serverless atualiza o que está disponível ECPUs/second no cache em 60 minutos e envia uma notificação de evento quando a atualização do limite mínimo é concluída.

Como funciona a pré-escalabilidade

Quando o limite mínimo ECPUs/second ou o armazenamento de dados são atualizados por meio do console, da CLI ou da API, esse novo limite fica disponível em 1 hora. ElastiCache O Serverless suporta 30K ECPUs/second em um cache vazio e até 90K ECPUs/sec ao usar o recurso Ler da réplica. ElastiCache O Serverless for Valkey 8.0 pode dobrar as solicitações suportadas por segundo (RPS) a cada 2-3 minutos, atingindo 5 milhões de RPS por cache de zero em menos de 13 minutos, com latência de leitura p50 consistente abaixo de um milissegundo. Se você prevê que um próximo evento de escalabilidade possa exceder essa taxa, recomendamos definir o mínimo ECPUs/second para o pico ECPUs/sec esperado pelo menos 60 minutos antes do evento de pico. Caso contrário, o aplicativo poderá apresentar latência elevada e limitação de solicitações.

Quando a atualização do limite mínimo for concluída, o ElastiCache Serverless começará a medir o novo mínimo ECPUs por segundo ou o novo armazenamento mínimo. Isso ocorre mesmo se o aplicativo não estiver executando solicitações no cache ou se o uso do armazenamento de dados estiver abaixo do mínimo. Quando você reduz o limite mínimo da configuração atual, a atualização é imediata, então o ElastiCache Serverless começará a medir o novo limite mínimo imediatamente.

 Note

- Ao definir um limite mínimo de uso, você é cobrado por esse limite, mesmo que seu uso real seja menor que o limite mínimo de uso. O uso de ECPU ou armazenamento de dados que exceda o limite mínimo de uso é cobrado de acordo com a taxa normal. Por exemplo, se você definir um limite mínimo de uso de 100.000 ECPUs/second, será cobrado pelo menos 1,224 USD por hora (usando os preços de ECPU em us-east-1), mesmo que seu uso seja inferior ao mínimo definido.
- ElastiCache O Serverless suporta a escala mínima solicitada em um nível agregado no cache. ElastiCache O Serverless também suporta um máximo de 30K ECPUs/second por slot (90K ECPUs/second ao usar Read from Replica usando conexões READONLY). Como prática recomendada, seu aplicativo deve garantir que a distribuição de chaves nos slots Valkey ou Redis OSS e o tráfego entre chaves sejam o mais uniformes possível.

Definindo limites de escala usando o console e AWS CLI

Definindo limites de escala usando o console AWS

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha o mecanismo em execução no cache que você deseja modificar.
3. É exibida uma lista de caches que executam o mecanismo escolhido.
4. Escolha o cache a ser modificado selecionando o botão de opção à esquerda do nome do cache.
5. Escolha Ações e Modificar.
6. Em Limites de uso, defina os limites de Memória ou Computar adequados.
7. Clique em Previsualizar alterações e em Salvar alterações.

Definindo limites de escala usando o AWS CLI

Para alterar os limites de escalabilidade usando a CLI, use `modify-serverless-cache` a 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}'
```

Remoção dos limites de escala usando a CLI

Para remover limites de escala usando a CLI, defina os parâmetros de limite mínimo e máximo como 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}'
```

Dimensionamento de clusters baseados em nós

A quantidade de dados que o seu aplicativo precisa processar é raramente estática. Ela aumenta e diminui à medida que sua empresa cresce ou passa por flutuações normais na demanda. Se você gerenciar automaticamente seu cache, você precisará provisionar hardware suficiente para seus picos de demanda, o que pode ser caro. Ao usar a Amazon, ElastiCache você pode escalar para atender à demanda atual, pagando somente pelo que você usa. ElastiCache permite que você escale seu cache para atender à demanda.

Note

Se um cluster Valkey ou Redis OSS for replicado em uma ou mais regiões, essas regiões serão escalonadas em ordem. Ao aumentar a escala verticalmente, as regiões secundárias

são dimensionadas primeiro e depois a região primária. Ao reduzir a escala verticalmente, a região primária é a primeira e, em seguida, todas as regiões secundárias. Ao atualizar a versão do mecanismo, a ordem é Região secundária e, em seguida, Região primária.

Tópicos

- [Escalar sob demanda clusters para o Memcached](#)
- [Escalabilidade manual para clusters do Memcached](#)
- [Escalonamento para Valkey ou Redis OSS \(modo cluster desabilitado\)](#)
- [Escarlar nós de réplica para Valkey ou Redis OSS \(modo cluster desativado\)](#)
- [Escalabilidade de clusters no Valkey ou Redis OSS \(modo cluster habilitado\)](#)

Escarlar sob demanda clusters para o Memcached

ElastiCache for Memcached oferece um serviço de cache em memória totalmente gerenciado que implanta, opera e escala verticalmente o Memcached na nuvem.AWS

Escalabilidade vertical sob demanda

Com o escalonamento vertical, o ElastiCache for Memcached fornece um sistema de cache de memória distribuído de alto desempenho, amplamente usado para acelerar aplicativos dinâmicos, aliviando a carga do banco de dados. Ele armazena dados e objetos na RAM, reduzindo a necessidade de leitura de fontes de dados externas.

Você pode aplicar a escalabilidade vertical aos clusters existentes baseados em nós, bem como aos novos. Essa ação pode proporcionar mais flexibilidade na alocação de recursos, permitindo que os usuários se adaptem com eficiência às mudanças nas workloads sem alterar a arquitetura do cluster. Essa capacidade de escalar melhora o desempenho aumentando a capacidade do cache durante os períodos de alta demanda e diminuindo a escala para otimizar os custos durante os períodos de baixa demanda. Isso simplifica as operações, elimina a necessidade de criar novos clusters para mudar os requisitos de recursos e permite respostas rápidas às flutuações do tráfego. No geral, a escalabilidade vertical para clusters baseados em nós do Memcached pode ajudar a aumentar a eficiência de custos, melhorar a utilização de recursos e até mesmo permitir que os usuários alterem o tipo de instância do Memcached. Tudo isso facilitando que os usuários alinhem sua infraestrutura de cache às necessidades reais do aplicativo.

 Note

- As modificações do tipo de nó só estão disponíveis para clusters do Memcached baseados em nós com as versões 1.5 ou posteriores do mecanismo.
- O Auto Discovery deve ser habilitado para que seja possível usar a escalabilidade vertical.

Configuração da escalabilidade vertical sob demanda para clusters do Memcached baseados em nós

Você pode configurar a escalabilidade vertical sob demanda para o Memcached com `scale-config`, que contém dois parâmetros:

1. `ScaleIntervalMinutes`: Tempo (em minutos) entre os lotes de escalabilidade durante o processo de atualização do Memcached
2. `ScalePercentage`: Porcentagem de nós a serem escalados simultaneamente durante o processo de atualização do Memcached

Convertendo um tipo de nó existente do Memcached em um cache que pode ser escalado verticalmente por meio da CLI

Para converter um cluster existente baseado em nós do Memcached em um cache que pode ser escalado verticalmente, você pode usar `elasticache modify-cache-cluster` por meio da CLI.

```
aws elasticache modify-cache-cluster \
  --cache-cluster-id <your-cluster-id> \
  --cache-node-type <new-node-type> \
  --scale-config <scale-config> \
  --apply-immediately
```

Configuração da escalabilidade vertical com a CLI

Para configurar a escalabilidade vertical para um cluster Memcached baseado em nós por meio da CLI, use `elasticache modify-cache-cluster` com `scale-config` e seus parâmetros `ScalePercentage` e `ScaleIntervalMinutes`.

- **scale-interval-minutes**: isso define o tempo (em minutos) entre os lotes de escalonamento. Essa configuração pode variar de 2 a 30 minutos. Se nenhum valor for especificado, será aplicado o valor padrão de 5 minutos.
- **porcentagem de escala**: especifica a porcentagem de nós a serem escalados simultaneamente em cada lote. Essa configuração pode variar de 10 a 100. A configuração é arredondada ao dividir. Por exemplo, se o resultado for 49,5, será aplicada uma configuração de 50. Se nenhum valor for especificado, será aplicado o valor padrão de 20.

Essas opções de configuração permitirão que você ajuste o processo de escalabilidade de acordo com suas necessidades específicas, equilibrando entre minimizar a interrupção do cluster e otimizar a velocidade de escalabilidade. O parâmetro `scale-config` será aplicável somente aos tipos de mecanismos Memcached e será ignorado para outros mecanismos de cache, garantindo compatibilidade com versões anteriores do uso atual da API em outros clusters.

Chamada de API

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id <your-cluster-id> \  
  --cache-node-type <new-node-type> \  
  --scale-config '{  
    "ScalePercentage": 30,  
    "ScaleIntervalMinutes": 2  
  }'  
  --apply-immediately
```

Resultado:

Retorna o ID do cluster e a alteração pendente.

```
{  
  "CacheCluster": {  
    "CacheNodeType": "old_instance_type",  
    ...  
    ...  
    "PendingModifiedValues": {  
      "CacheNodeType": "new_instance_type"  
    },  
  },  
}
```

Listar sua configuração de escalabilidade vertical do cache do Memcached

Você pode recuperar as opções de escalabilidade para os caches do Memcached e ver quais são as opções atuais para a escalabilidade vertical.

Chamada de API

```
aws elasticache list-allowed-node-type-modifications --cache-cluster-id <your-cluster-id>
```

Resultado:

```
{
  "ScaleUpModifications": [
    "cache.x.xxxx",
    "cache.x.xxxx"
  ],
  "ScaleDownModifications": [
    "cache.x.xxxx",
    "cache.x.xxxx",
    "cache.x.xxxx"
  ]
}
```

Dimensionamento vertical para Memcached com o Console de gerenciamento da AWS

Siga estas etapas para usar o Console de gerenciamento da AWS para converter um cluster Memcached baseado em nós em um cluster escalável verticalmente.

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Selecione o cluster do Memcached a ser convertido.
3. Selecione a guia Modify (Modificar).
4. Vá até a seção Configurações de cache e selecione o tipo de nó desejado.
5. Selecione Visualizar alterações e revise as alterações.
6. Selecione Modify.

Escalabilidade horizontal automatizada para Memcached

ElastiCache agora se integra ao serviço AWS Application Auto Scaling (AAS) para incluir escalabilidade horizontal automatizada para clusters Memcached. Você pode definir políticas de escalabilidade por meio do serviço AWS Application Auto Scaling e ajustar automaticamente o número de nós nos clusters do Memcached conforme necessário, com base em métricas ou cronogramas predefinidos.

Note

No momento, a escalabilidade horizontal automatizada não está disponível nas regiões Pequim e Ningxia.

Esses são os métodos disponíveis para escalar horizontal e automaticamente seus clusters baseados em nós.

- **Escalabilidade programada:** a escalabilidade com base em uma programação permite que você defina sua própria programação de escalabilidade para alterações de carga previsíveis. Por exemplo, toda semana o tráfego para seu aplicativo web começa a aumentar na quarta-feira, permanece alto na quinta-feira e começa a diminuir na sexta-feira. É possível configurar o recurso Auto Scaling para aumentar a capacidade na quarta-feira e diminuir a capacidade na sexta-feira.
- **Monitoramento do objetivo:** com as políticas de dimensionamento com monitoramento do objetivo, você seleciona uma métrica de escalabilidade e define um valor pretendido. O Application Auto Scaling cria e gerencia os CloudWatch alarmes que acionam a política de escalabilidade e calcula o ajuste de escalabilidade com base na métrica e no valor alvo. A política de escalabilidade adiciona ou remove capacidade conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele.

Como configurar a escalabilidade horizontal para um cluster Memcached baseado em nós por meio da CLI

Ao escalar horizontalmente um cluster do Memcached baseado em nós, você pode usar uma política de monitoramento do objetivo, uma política programada ou ambas.

1. Registre um recurso como alvo escalável

Chame a `RegisterScalableTarget` API no AWS Application Auto Scaling para registrar o destino da dimensão escalável. `elasticache:cache-cluster:Nodes`

API: `ApplicationAutoScaling.RegisterScalableTarget`

Entrada:

```
{
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ResourceId": "cache-cluster/test-cluster-1",
  "ServiceNamespace": "elasticache",
  "MinCapacity": 20,
  "MaxCapacity": 50
}
```

2. Criar uma política de escalabilidade com monitoramento do objetivo

Em seguida, você pode criar uma política de escalabilidade de rastreamento do objetivo para o recurso chamando a API `put scaling policy`.

3. Métrica pré-definida

A seguir está uma política que se expande de acordo com a dimensão do nó de cache, usando a métrica predefinida `ElastiCacheCPUUtilization`, definida como 50 para o cluster `test-cluster-1`. Ao excluir nós para aumentar a escala, os últimos `n` nós serão removidos.

API: `ApplicationAutoScaling.PutScalingPolicy`

Entrada:

```
{
  "PolicyName": "cpu50-target-tracking-scaling-policy",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 50,
    "PredefinedMetricSpecification": {
      "PredefinedMetricType": "ElastiCacheCPUUtilization"
    },
    "ScaleOutCooldown": 600,
    "ScaleInCooldown": 600
  },
  "ServiceNamespace": "elasticache",
}
```

```
"ScalableDimension": "elasticache:cache-cluster:Nodes",
"ResourceId": "cache-cluster/test-cluster-1"
}
```

Saída:

```
{
  "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/elasticache/cache-cluster/test-cluster-1:policyName/cpu50-target-tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca",
      "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
    },
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d",
      "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
    }
  ]
}
```

4. Métrica personalizada

Também é possível definir uma política de escalabilidade por dimensão usando uma porcentagem personalizada baseada na métrica do Cloudwatch.

Entrada:

```
{
  "PolicyName": "cpu50-target-tracking-scaling-policy",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "CustomizedMetricSpecification": {
      "Dimensions": [
        {
          "Name": "MyMetricDimension",
```

```
    "Value": "DimensionValue"
  },
  ],
  "MetricName": "MyCustomMetric",
  "Namespace": "MyNamespace",
  "Statistic": "Average",
  "Unit": "Percent"
},
"TargetValue": 40,
"ScaleOutCooldown": 600,
"ScaleInCooldown": 600
},
"ServiceNamespace": "elasticache",
"ScalableDimension": "elasticache:cache-cluster:Nodes",
"ResourceId": "cache-cluster/test-cluster-1"
}
```

5. Ações programadas

Quando precisar aumentar a escala horizontalmente para um evento específico e depois reduzir a escala horizontalmente depois do evento, você pode criar duas ações agendadas chamando a API `PutScheduledAction`.

Política 1: aumentar a escala horizontalmente

O comando `at` em `--schedule` programa a ação para ser executada uma vez em uma data e hora especificadas no futuro. O campo de programação também é compatível com taxa (minuto, hora, dia etc.) e cron (para expressão cron).

Na data e hora especificadas, o Application Auto Scaling atualiza os valores `MinCapacity` e `MaxCapacity`. O Application Auto Scaling se expande `MinCapacity` para colocar os nós de cache em 70.

API: `ApplicationAutoScaling.PutScheduledAction`

Entrada:

```
{
  "ResourceId": "elasticache:ache-cluster:test-cluster-1",
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ScalableTargetAction": {
    "MaxCapacity": 100,
```

```
"MinCapacity": 70
},
"Schedule": "at(2020-05-20T17:05:00)",
"ScheduledActionName": "ScalingOutScheduledAction",
"ServiceNamespace": "elasticache",
}
```

Política 2: Reduzir a escala horizontalmente

Na data e hora especificadas, o Application Auto Scaling atualiza a `MinCapacity` e a `MaxCapacity` da tabela e reduz a escala na horizontal para `MaxCapacity`, para retornar os nós de cache para 60.

API: `ApplicationAutoScaling.PutScheduledAction`

Entrada:

```
{
  "ResourceId": "elasticache:cache-cluster:test-cluster-1",
  "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ScalableTargetAction": {
    "MaxCapacity": 60,
    "MinCapacity": 40
  },
  "Schedule": "at(2020-05-21T17:05:00)",
  "ScheduledActionName": "ScalingInScheduledAction",
  "ServiceNamespace": "elasticache",
}
```

6. Visualizar as atividades de escalabilidade

Você pode visualizar as atividades de escalabilidade usando a API `DescribeScalingActivities`.

API: `ApplicationAutoScaling.DescribeScalingActivities`

Saída:

```
{
  "ScalingActivities": [
    {
      "ScalableDimension": "elasticache:elasticache:DesiredCount",
```

```

    "Description": "Setting desired count to 30.",
    "ResourceId": "elasticache/cache-cluster/test-cluster-1",
    "ActivityId": "4d759079-a31f-4d0c-8468-504c56e2eecf",
    "StartTime": 1462574194.658,
    "elasticacheNamespace": "elasticache",
    "EndTime": 1462574276.686,
    "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy cpu50-target-tracking-scaling-policy",
    "StatusMessage": "Failed to set desired count to 30",
    "StatusCode": "Failed"
  },
  {
    "ScalableDimension": "elasticache:elasticache:DesiredCount",
    "Description": "Setting desired count to 25.",
    "ResourceId": "elasticache/cache-cluster/test-cluster-1",
    "ActivityId": "90aff0eb-dd6a-443c-889b-b809e78061c1",
    "StartTime": 1462574254.223,
    "elasticacheNamespace": "elasticache",
    "EndTime": 1462574333.492,
    "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy cpu50-target-tracking-scaling-policy",
    "StatusMessage": "Successfully set desired count to 25. Change successfully fulfilled by elasticache.",
    "StatusCode": "Successful"
  }
]
}

```

7. Editar/excluir política de escalabilidade

Você pode editar ou excluir políticas chamando a API `PutScalingPolicy` novamente ou chamando a ação `DeleteScalingPolicy` ou `DeleteScheduled`.

8. Cancelar registro de destinos dimensionáveis

Você pode cancelar o registro de destinos dimensionáveis por meio da API `DeregisterScalableTarget`. O cancelamento do registro de um destino dimensionável exclui as políticas de escalabilidade e as ações programadas associadas a ele.

API: `ApplicationAutoScaling`. `DeregisterScalableTarget`

Entrada:


```
{
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ServiceNamespace": "elasticache",
  "ScalableDimension": "elasticache:cache-cluster:Nodes"
}
```

9. Limpeza de políticas de escalabilidade

10. Várias políticas de escalabilidade

Você pode criar várias políticas de escalabilidade. A seguir estão os principais textos explicativos sobre o comportamento do [rastreamento de destinos do Auto Scaling](#).

- Você pode ter várias políticas de escalabilidade de rastreamento de destino para um destino escalável, desde que cada uma delas use uma métrica diferente.
- A intenção do Application Auto Scaling é sempre priorizar a disponibilidade, portanto, seu comportamento será diferente dependendo se as políticas de monitoramento do objetivo estão prontas para aumentar ou reduzir a escala. Ele vai expandir o destino dimensionável se qualquer uma das políticas de rastreamento de destino estiverem prontas para expandir, mas vai reduzir somente se todas as políticas de rastreamento de destino (com a parte de redução habilitada) estiverem prontas para reduzir
- Se várias políticas instruírem o destino escalável a expandir ou reduzir ao mesmo tempo, o Application Auto Scaling fará a escalabilidade com base na política que fornece a maior capacidade tanto para expandir como para reduzir. Isso oferece maior flexibilidade para cobrir vários cenários e garante que sempre haja capacidade suficiente para processar suas cargas de trabalho de aplicativos.

Note

AWS O Application Auto Scaling não coloca políticas de escalabilidade em fila. O Application Auto Scaling aguardará a conclusão do primeiro escalonamento, depois esfriará e repetirá o algoritmo acima.

Dimensione horizontalmente automaticamente um cluster Memcached baseado em nós por meio do Console de gerenciamento da AWS

Siga estas etapas para usar o Console de gerenciamento da AWS para converter um cluster Memcached existente baseado em nós em um cluster escalável horizontalmente.

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Selecione o cache do Memcached a ser convertido.
3. Acesse a guia Autoscaling (Escalonamento automático).
4. Selecione a política de escalabilidade a ser aplicada selecionando Add dynamic scaling (Adicionar escalabilidade dinâmica) ou Add scheduled scaling (Adicionar escalabilidade programada).
5. Preencha os detalhes da política selecionada conforme o necessário.
6. Clique em Criar.

Escalabilidade manual para clusters do Memcached

Escalar horizontal ou verticalmente um cluster do Memcached é tão fácil quanto adicionar ou remover nós do cluster. Os clusters do Memcached são compostos de 1 a 60 nós.

Como você pode particionar seus dados em todos os nós em um cluster Memcached, raramente é necessário expandir para um tipo de nó com mais memória. No entanto, como o mecanismo Memcached não mantém a persistência de dados, se você dimensionar um tipo de nó diferente, seu novo cluster Memcached começará vazio, a menos que seu aplicativo o preencha.

Crie um novo cluster para escalabilidade manual e vertical do cluster do Memcached. Os clusters Memcached sempre começam vazios, a menos que seu aplicativo o preencha.

Escalar manualmente clusters do Memcached

Ação	Tópico
Aumento de escala	Adição de nós a um cluster
Redução de escala	Exclusão de nós de um cluster
Alteração nos tipos de nó	Escalar manual e verticalmente clusters do Memcached baseados em nós

Tópicos

- [Escalar manualmente um cluster do Memcached baseado em nós horizontalmente](#)
- [Escalar manual e verticalmente clusters do Memcached baseados em nós](#)

Escalar manualmente um cluster do Memcached baseado em nós horizontalmente

O mecanismo Memcached oferece suporte para particionar seus dados em vários nós. Por isso, os clusters Memcached escalam horizontalmente com facilidade. Para dimensionar horizontalmente seu cluster Memcached, basta adicionar ou remover nós.

Os tópicos a seguir detalham como escalar seu cluster Memcached inserindo ou removendo nós.

- [Adição de nós a um cluster](#)
- [Exclusão de nós do cluster](#)

Cada vez que você altera o número de nós no seu cluster Memcached, você deve remapear pelo menos alguns dos seus espaços de chaves para que ele seja mapeado para o nó correto. Para obter informações mais detalhadas sobre o balanceamento de carga do cluster Memcached, consulte [Configurando seu ElastiCache cliente para balanceamento de carga eficiente \(Memcached\)](#).

Se você usar a descoberta automática no seu cluster Memcached, não precisará alterar os endpoints em seu aplicativo à medida que você adiciona ou remove nós. Para obter mais informações sobre a descoberta automática, consulte [Identificar automaticamente nós no seu cluster \(Memcached\)](#). Se você não usar a descoberta automática, cada vez que você alterar o número de nós no seu cluster Memcached, deverá atualizar os endpoints no seu aplicativo.

Escalar manual e verticalmente clusters do Memcached baseados em nós

Quando você escala ou reduz seu cluster Memcached manualmente, deve criar um novo cluster. Os clusters Memcached sempre começam vazios, a menos que seu aplicativo o preencha.

Important

Se você estiver dimensionando para um tipo de nó menor, certifique-se de que o tipo de nó menor seja adequado para seus dados e despesas gerais. Para obter mais informações, consulte [Escolha do tamanho do nó](#).

Tópicos

- [Escalar verticalmente um cluster do Memcached baseado em nós \(console\)](#)
- [Escalar verticalmente um cluster do Memcached baseado em nós \(AWS CLI\)](#)
- [Dimensionando verticalmente um cluster Memcached baseado em nós \(API\) ElastiCache](#)

Escalar verticalmente um cluster do Memcached baseado em nós (console)

O procedimento a seguir demonstra passo a passo como escalar um cluster do Memcached baseado em nós verticalmente usando o Console de gerenciamento da AWS.

1. Crie um novo cluster com o novo tipo de nó. Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).
2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para obter mais informações, consulte [Localização de endpoints de um cluster \(console\) \(Memcached\)](#).
3. Exclua o cluster antigo. Para obter mais informações, consulte [Deleting a new node in Memcached](#).

Escalar verticalmente um cluster do Memcached baseado em nós (AWS CLI)

O procedimento a seguir demonstra passo a passo como escalar um cluster do Memcached baseado em nós verticalmente usando o AWS CLI.

1. Crie um novo cluster com o novo tipo de nó. Para obter mais informações, consulte [Criar um cluster \(AWS CLI\)](#).
2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para obter mais informações, consulte [Localizar endpoints \(AWS CLI\)](#).
3. Exclua o cluster antigo. Para obter mais informações, consulte [Usando o AWS CLI para excluir um ElastiCache cluster](#).

Dimensionando verticalmente um cluster Memcached baseado em nós (API) ElastiCache

O procedimento a seguir mostra como escalar verticalmente um cluster Memcached baseado em nós usando a API. ElastiCache

1. Crie um novo cluster com o novo tipo de nó. Para obter mais informações, consulte [Criação de um cluster para Memcached \(API\) ElastiCache](#).

2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para obter mais informações, consulte [Localização de endpoints \(API do ElastiCache\)](#).
3. Exclua o cluster antigo. Para obter mais informações, consulte [Usando a ElastiCache API](#).

Escalonamento para Valkey ou Redis OSS (modo cluster desabilitado)

Os clusters do Valkey ou Redis OSS (modo cluster desabilitado) podem ser um cluster de nó único com 0 fragmentos ou clusters de vários nós com 1 fragmento. Os clusters de nó único usam esse único nó para leituras e gravações. Clusters de vários nós sempre têm 1 nó como nó read/write principal com 0 a 5 nós de réplica somente para leitura.

Tópicos

- [Escalonamento para Valkey ou Redis OSS \(modo cluster desabilitado\)](#)

Escalonamento de clusters Valkey ou Redis OSS

Ação	Valkey ou Redis OSS (modo cluster desabilitado)	Valkey ou Redis OSS (modo cluster habilitado)
Redução de escala	Removendo nós de um ElastiCache cluster	Escalabilidade de clusters no Valkey ou Redis OSS (modo cluster habilitado)
Aumento de escala	Adição de nós a um cluster	Refragmentação online para o Valkey ou o Redis OSS (modo cluster habilitado)
Alteração nos tipos de nó	Para um tipo de nó maior: • Aumento de escala vertical de clusters do Valkey ou Redis OSS de nó único • Aumento de escala vertical de clusters do Valkey ou Redis OSS com réplicas Para um tipo de nó menor: • Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único	Escalabilidade vertical online com modificação do tipo de nó

Ação	Valkey ou Redis OSS (modo cluster desabilitado)	Valkey ou Redis OSS (modo cluster habilitado)
	Redução de escala vertical de clusters do Valkey ou Redis OSS com réplicas	
Alteração do número de grupos de nós	Não há suporte para clusters do Valkey ou Redis OSS (modo cluster desabilitado)	Escalabilidade de clusters no Valkey ou Redis OSS (modo cluster habilitado)

Sumário

- [Escalonamento para Valkey ou Redis OSS \(modo cluster desabilitado\)](#)
 - [Aumento de escala vertical de clusters do Valkey ou Redis OSS de nó único](#)
 - [Aumento de escala vertical de clusters de nó único do Valkey ou Redis OSS \(modo cluster desabilitado\) \(console\)](#)
 - [Aumento de escala vertical de clusters do Valkey ou Redis OSS de nó único \(AWS CLI\)](#)
 - [Escalando clusters OSS \(API\) Valkey ou Redis de nó único ElastiCache](#)
 - [Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único](#)
 - [Redução de escala vertical de um cluster do Valkey ou Redis OSS de nó único \(console\)](#)
 - [Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único \(AWS CLI\)](#)
 - [Reduzindo a escala de clusters Valkey ou Redis OSS \(API\) de nó único ElastiCache](#)

Escalonamento para Valkey ou Redis OSS (modo cluster desabilitado)

Nós do Valkey ou Redis OSS (modo cluster desabilitado) devem ser suficientemente grandes para conter todos os dados do cache mais a sobrecarga do Valkey ou Redis OSS. Para alterar a capacidade de dados do seu cluster do Valkey ou Redis OSS (modo cluster desabilitado), você deve escalar verticalmente, aumentando a escala verticalmente para um tipo de nó maior para aumentar a capacidade de dados ou reduzindo para um tipo de nó menor para diminuir a capacidade de dados.

O processo ElastiCache de escalabilidade foi projetado para fazer o melhor esforço para reter seus dados existentes e requer uma replicação bem-sucedida do Valkey ou do Redis OSS. Para clusters do Valkey ou Redis OSS (modo de cluster desabilitado), recomendamos que seja disponibilizada memória suficiente para o Valkey ou Redis OSS.

Não é possível particionar seus dados em vários clusters Valkey ou Redis OSS (modo cluster desabilitado). No entanto, se você só precisa aumentar ou diminuir a capacidade de leitura do seu cluster, pode criar um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com nós de réplica e adicionar ou remover réplicas de leitura. Para criar um cluster do Valkey ou Redis OSS (modo cluster desabilitado) com nós de réplica usando seu cluster Valkey ou Redis OSS de um único nó como cluster primário, consulte [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#).

Depois de criar o cluster com réplicas, você pode aumentar a capacidade de leitura adicionando réplicas de leitura. Mais tarde, se precisar, você pode reduzir a capacidade de leitura removendo réplicas de leitura. Para acessar mais informações, consulte [Aumento da capacidade de leitura](#) ou [Redução da capacidade de leitura](#).

Além de poderem escalonar a capacidade de leitura, clusters do Valkey ou Redis OSS (modo cluster desabilitado) com réplicas fornecem outras vantagens comerciais. Para obter mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Valkey ou Redis OSS, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster.

Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso.

Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

Tópicos

- [Aumento de escala vertical de clusters do Valkey ou Redis OSS de nó único](#)
- [Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único](#)

Aumento de escala vertical de clusters do Valkey ou Redis OSS de nó único

Ao escalar um cluster Valkey ou Redis OSS de nó único, ElastiCache executa o processo a seguir, independentemente de você usar o ElastiCache console, o ou a API AWS CLI. ElastiCache

1. Um novo cluster com o novo tipo de nó é ativado na mesma zona de disponibilidade que o cluster existente.
2. Os dados do cache no cluster existente são copiados para o novo cluster. A duração desse processo depende do tipo de nó e da quantidade de dados no cluster.
3. As leituras e as gravações são atendidas com o novo cluster. Como os endpoints do novo cluster são os mesmos do que para o cluster antigo, você não precisa atualizar os endpoints no seu aplicativo. Você perceberá uma breve interrupção (alguns segundos) das leituras e gravações no nó primário enquanto a entrada do DNS for atualizada.
4. ElastiCache exclui o cluster antigo. Você perceberá uma breve interrupção (alguns segundos) das leituras e gravações no nó antigo porque as conexões ao nó antigo serão desconectadas.

Note

Para clusters que executam o tipo de nó r6gd, você só pode escalar para tamanhos de nó dentro da família de nós r6gd.

Conforme mostrado na tabela a seguir, sua operação de expansão do Valkey ou Redis OSS será bloqueada se você tiver uma atualização do mecanismo programada para a próxima janela de manutenção. Para obter mais informações sobre janelas de manutenção, consulte [Gerenciamento de manutenção de cluster do ElastiCache](#).

Operações bloqueadas do Valkey ou do Redis OSS

Operações pendentes	Operações bloqueadas
Amplie a sua capacidade	Atualização imediata do mecanismo
Atualização do mecanismo	Expansão imediata
Expansão e atualização do mecanismo	Expansão imediata
	Atualização imediata do mecanismo

Se você tiver uma operação pendente que está bloqueando você, será possível tomar uma destas providências.

- Planeje sua operação de expansão do Valkey ou Redis OSS para a próxima janela de manutenção, desmarcando a caixa de seleção Apply immediately (CLI use: `--no-apply-immediately`, API use: `ApplyImmediately=false`).
- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de expansão do Valkey ou Redis OSS.
- Adicione a atualização do mecanismo Valkey ou Redis OSS a essa modificação de cluster com a caixa de seleção Apply Immediately (Aplicar imediatamente) marcada (CLI use: `--apply-immediately`, API use: `ApplyImmediately=true`). Isso desbloqueia sua operação de expansão, fazendo com que a atualização do mecanismo seja realizada imediatamente.

Você pode escalar um cluster Valkey ou Redis OSS de nó único (modo de cluster desativado) usando o ElastiCache console, o ou a AWS CLI API. ElastiCache

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Valkey ou Redis OSS, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster.

Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso.

Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

Aumento de escala vertical de clusters de nó único do Valkey ou Redis OSS (modo cluster desabilitado) (console)

O procedimento a seguir descreve como escalar um cluster OSS Valkey ou Redis de nó único usando o Management Console. ElastiCache Durante esse processo, o cluster do Valkey ou Redis OSS continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala vertical de um cluster Valkey ou Redis OSS de nó único (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey or Redis OSS clusters (Clusters do Valkey ou Redis OSS).
3. Na lista de clusters, escolha o cluster que você deseja expandir (ele deve estar executando o mecanismo Valkey ou Redis OSS, e não o mecanismo clustered Valkey ou Redis OSS).
4. Escolha Modificar.
5. No assistente Modify Cluster:
 - a. Na lista Node type, escolha o tipo de nó a partir do qual você deseja escalar.
 - b. Se você estiver usando `reserved-memory` para gerenciar sua memória, na lista Parameter Group, escolha o parameter group personalizado que reserva a quantidade correta de memória para o seu novo tipo de nó.
6. Se quiser executar o processo de expansão imediatamente, escolha a caixa `Apply immediately`. Se a caixa `Apply immediately` não for escolhida, o processo de expansão será realizado durante a próxima janela de manutenção desse cluster.
7. Escolha Modificar.

Se você escolheu `Apply immediately` na etapa anterior, o status do cluster mudará para `modifying`. Quando o status mudar para `available`, a modificação estará completa, e você poderá começar a usar o novo cluster.

Aumento de escala vertical de clusters do Valkey ou Redis OSS de nó único (AWS CLI)

O procedimento a seguir descreve como expandir um cluster Valkey ou Redis OSS de um único nó usando a AWS CLI. Durante esse processo, o cluster do Valkey ou Redis OSS continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala verticalmente de um cluster Valkey ou Redis OSS de um único nó (AWS CLI)

1. Determine os tipos de nós para os quais você pode escalar executando o `AWS CLI list-allowed-node-type-modifications` comando com o parâmetro a seguir.
 - `--cache-cluster-id`

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ],  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small"  
  ],  
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu cluster existente especificando o cluster a ser expandido e o novo tipo de nó maior, usando o `AWS CLI modify-cache-cluster` comando e os parâmetros a seguir.
 - `--cache-cluster-id`: o nome do cluster que você está aumentando.
 - `--cache-node-type`: o novo tipo de nó para o qual você deseja escalar o cluster. Esse valor deve ser um dos tipos de nós retornados pelo comando `list-allowed-node-type-modifications` na etapa 1.
 - `--cache-parameter-group-name` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
 - `--apply-immediately`: faz com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção do cluster, use o parâmetro `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cache-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m2-xl \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-redis-cache-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m2-xl ^  
  --apply-immediately
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "CacheCluster": {  
    "Engine": "redis",  
    "CacheParameterGroup": {
```

```

    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.redis6.x",
    "ParameterApplyStatus": "in-sync"
  },
  "SnapshotRetentionLimit": 1,
  "CacheClusterId": "my-redis-cache-cluster",
  "CacheSecurityGroups": [],
  "NumCacheNodes": 1,
  "SnapshotWindow": "00:00-01:00",
  "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
  "AutoMinorVersionUpgrade": true,
  "CacheClusterStatus": "modifying",
  "PreferredAvailabilityZone": "us-west-2a",
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
  "CacheSubnetGroupName": "default",
  "EngineVersion": "6.0",
  "PendingModifiedValues": {
    "CacheNodeType": "cache.m3.2xlarge"
  },
  "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
  "CacheNodeType": "cache.m3.medium",
  "DataTiering": "disabled"
}
}

```

Para obter mais informações, consulte [modify-cache-cluster](#) na AWS CLI Referência.

3. Se você usou o `--apply-immediately`, verifique o status do novo cluster usando o AWS CLI `describe-cache-clusters` comando com o parâmetro a seguir. Quando o status mudar para `available`, você poderá começar a usar o novo cluster maior.
 - `--cache-cluster-id`: o nome do cluster do Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster específico em vez de todos os clusters.

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

Para obter mais informações, consulte [describe-cache-clusters](#) na AWS CLI Referência.

Escalando clusters OSS (API) Valkey ou Redis de nó único ElastiCache

O procedimento a seguir descreve como escalar um cluster OSS Valkey ou Redis de nó único usando a API. ElastiCache Durante esse processo, o cluster do Valkey ou Redis OSS continuará a atender solicitações com tempo de inatividade mínimo.

Para escalar um cluster OSS (API) Valkey ou Redis de nó único ElastiCache

1. Determine os tipos de nós para os quais você pode escalar executando a `ListAllowedNodeTypeModifications` ação da ElastiCache API com o parâmetro a seguir.
 - `CacheClusterId`: o nome do cluster Valkey ou Redis OSS de nó único que você deseja aumentar a escala verticalmente.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) a Amazon ElastiCache API Reference.

2. Modifique seu cluster existente especificando o cluster a ser expandido e o novo tipo de nó maior, usando a ação da `ModifyCacheCluster` ElastiCache API e os parâmetros a seguir.
 - `CacheClusterId`: o nome do cluster que você está aumentando.
 - `CacheNodeType`: o novo tipo de nó maior para o qual você deseja aumentar a escala vertical do cluster. Esse valor deve ser um dos tipos de nós retornados pela ação `ListAllowedNodeTypeModifications` na etapa anterior.
 - `CacheParameterGroupName` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.

- **ApplyImmediately**: defina como `true` para fazer com que o processo de expansão seja executado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção do cluster, use `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &CacheClusterId=MyRedisCacheCluster  
  &CacheNodeType=cache.m3.xlarge  
  &CacheParameterGroupName redis32-m2-x1  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyCacheCluster](#) a Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, verifique o status do novo cluster usando a `DescribeCacheClusters` ação da ElastiCache API com o parâmetro a seguir. Quando o status mudar para `available`, você poderá começar a usar o novo cluster maior.
- **CacheClusterId**: o nome do cluster do Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster específico em vez de todos os clusters.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=MyRedisCacheCluster  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeCacheClusters](#) a Amazon ElastiCache API Reference.

Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único

As seções a seguir demonstram como reduzir um cluster Valkey ou Redis OSS de nó único para um tipo de nó menor. Garantir que o novo tipo de nó menor seja suficientemente grande para acomodar todos os dados e a sobrecarga do Valkey ou Redis OSS é importante para o sucesso a longo prazo do seu novo cluster Valkey ou Redis OSS. Para obter mais informações, consulte [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#).

Note

Para clusters que executam o tipo de nó r6gd, você só pode escalar para tamanhos de nó dentro da família de nós r6gd.

Tópicos

- [Redução de escala vertical de um cluster do Valkey ou Redis OSS de nó único \(console\)](#)
- [Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único \(AWS CLI\)](#)
- [Reduzindo a escala de clusters Valkey ou Redis OSS \(API\) de nó único ElastiCache](#)

Redução de escala vertical de um cluster do Valkey ou Redis OSS de nó único (console)

O procedimento a seguir mostra como escalar seu cluster OSS Valkey ou Redis de nó único para um tipo de nó menor usando o console. ElastiCache

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Valkey ou Redis OSS, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster.

Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso.

Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

Para reduzir a escala vertical do seu cluster do Valkey ou Redis OSS de nó único (console)

1. Certifique-se de que o tipo de nó menor seja adequado para suas necessidades de dados e sobrecarga.
2. Se o seu parameter group usar `reserved-memory` para reservar memória para a sobrecarga do Valkey ou Redis OSS, certifique-se de ter um parameter group personalizado para reservar a quantidade correta de memória para o seu novo tipo de nó.

Alternativamente, você pode modificar seu parameter group personalizado para usar `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

3. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
4. Na lista de clusters, escolha o cluster que você deseja reduzir. Esse cluster deve estar executando o mecanismo Valkey ou Redis OSS, e não o mecanismo clustered Valkey ou Redis OSS.
5. Escolha Modificar.
6. No assistente Modify Cluster:
 - a. Na lista Node type (Tipo de nó), escolha o tipo de nó para o qual deseja reduzir.
 - b. Se você estiver usando `reserved-memory` para gerenciar sua memória, na lista Parameter Group, escolha o parameter group personalizado que reserva a quantidade correta de memória para o seu novo tipo de nó.
7. Se desejar executar o processo de redução imediatamente, escolha a caixa de seleção Apply immediately (Aplicar imediatamente). Se a caixa de seleção Apply immediately (Aplicar imediatamente) não for escolhida, o processo de redução será realizado durante a próxima janela de manutenção desse cluster.
8. Escolha Modificar.
9. Quando o status do cluster muda de modifying para available, significa que o seu cluster foi escalonado para o novo tipo de nó. Não é necessário atualizar os endpoints no seu aplicativo.

Redução de escala vertical de clusters do Valkey ou Redis OSS de nó único (AWS CLI)

O procedimento a seguir descreve como reduzir um cluster Valkey ou Redis OSS de um único nó usando a AWS CLI.

Para reduzir a escala verticalmente de um cluster Valkey ou Redis OSS de um único nó (AWS CLI)

1. Determine os tipos de nós para os quais você pode reduzir executando o AWS CLI `list-allowed-node-type-modifications` comando com o parâmetro a seguir.

- `--cache-cluster-id`

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ],  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small",  
  ],  
}
```

```
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu cluster existente especificando o cluster a ser reduzido e o novo tipo de nó menor, usando o `AWS CLI modify-cache-cluster` comando e os parâmetros a seguir.
 - `--cache-cluster-id`: o nome do cluster para o qual você está reduzindo a escala verticalmente.
 - `--cache-node-type`: o novo tipo de nó para o qual você deseja escalar o cluster. Esse valor deve ser um dos tipos de nós retornados pelo comando `list-allowed-node-type-modifications` na etapa 1.
 - `--cache-parameter-group-name` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
 - `--apply-immediately`: faz com que o processo de redução da escala vertical seja aplicado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção do cluster, use o parâmetro `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cache-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m2-xl \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-redis-cache-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m2-xl ^  
  --apply-immediately
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{
  "CacheCluster": {
    "Engine": "redis",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.redis6.x",
      "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
      "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
  }
}
```

Para obter mais informações, consulte [modify-cache-cluster](#) na AWS CLI Referência.

3. Se você usou `--apply-immediately`, verifique o status do novo cluster usando o `AWS CLI describe-cache-clusters` comando com o parâmetro a seguir. Quando o status mudar para `available`, você poderá começar a usar o novo cluster maior.
- `--cache-cluster-id`: o nome do cluster do Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster específico em vez de todos os clusters.

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

Para obter mais informações, consulte [describe-cache-clusters](#) na AWS CLI Referência.

Reduzindo a escala de clusters Valkey ou Redis OSS (API) de nó único ElastiCache

O procedimento a seguir descreve como aumentar a escala de um cluster Valkey ou Redis OSS de nó único usando a API. ElastiCache

Para reduzir um cluster OSS (API) Valkey ou Redis de nó único ElastiCache

1. Determine os tipos de nós para os quais você pode reduzir executando a `ListAllowedNodeTypeModifications` ação da ElastiCache API com o parâmetro a seguir.
 - `CacheClusterId`: o nome do cluster de cache do Valkey ou Redis OSS de nó único que você deseja reduzir a escala vertical.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) a Amazon ElastiCache API Reference.

2. Modifique seu cluster existente especificando o cluster a ser expandido e o novo tipo de nó maior, usando a ação da `ModifyCacheCluster` ElastiCache API e os parâmetros a seguir.
 - `CacheClusterId`: o nome do cluster para o qual você está reduzindo a escala verticalmente.
 - `CacheNodeType`: o novo tipo de nó menor para o qual você deseja reduzir a escala vertical do cluster. Esse valor deve ser um dos tipos de nós retornados pela ação `ListAllowedNodeTypeModifications` na etapa anterior.

- **CacheParameterGroupName** – [Opcional] Use esse parâmetro se estiver usando **reserved-memory** para gerenciar a memória reservada do cluster. Especifique um **parameter group** de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando **reserved-memory-percent**, você poderá omitir esse parâmetro.
- **ApplyImmediately**: defina como **true** para fazer com que o processo de redução de escala vertical seja executado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção do cluster, use **ApplyImmediately=false**.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=true  
&CacheClusterId=MyRedisCacheCluster  
&CacheNodeType=cache.m3.xlarge  
&CacheParameterGroupName=redis32-m2-xl  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyCacheCluster](#) a Amazon ElastiCache API Reference.

3. Se você usou **ApplyImmediately=true**, verifique o status do novo cluster usando a **DescribeCacheClusters** ação da ElastiCache API com o parâmetro a seguir. Quando o status mudar para **available** (disponível), você poderá começar a usar o novo cluster menor.
- **CacheClusterId**: o nome do cluster do Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster específico em vez de todos os clusters.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeCacheClusters](#) a Amazon ElastiCache API Reference.

Escalar nós de réplica para Valkey ou Redis OSS (modo cluster desativado)

Um cluster Valkey ou Redis OSS com nós de réplica (chamado de grupo de replicação na API/CLI) fornece alta disponibilidade por meio de uma replicação que possui o recurso Multi-AZ com failover automático habilitado. Um cluster com nós de réplica é um conjunto lógico de até seis clusters Valkey ou Redis OSS em que um nó, o primário, pode atender a solicitações de leitura e gravação. Todos os outros nós do cluster são réplicas somente leitura do primário. Os dados escritos no primário são replicados de forma assíncrona para todas as réplicas de leitura no cluster. Como o Valkey ou Redis OSS (modo cluster desabilitado) não oferece suporte ao particionamento dos seus dados em vários clusters, cada nó em um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) contém todo o conjunto de dados do cache. Clusters do Valkey ou Redis OSS (modo cluster habilitado) oferecem suporte para o particionamento dos seus dados até 500 fragmentos.

Para alterar a capacidade de dados do seu cluster, você deve expandi-lo para um tipo de nó maior ou reduzi-lo para um tipo de nó menor.

Para alterar a capacidade de leitura do seu cluster, adicione mais réplicas de leitura, até um máximo de 5, ou remova réplicas de leitura.

O processo ElastiCache de escalabilidade foi projetado para fazer o melhor esforço para reter seus dados existentes e requer uma replicação bem-sucedida do Valkey ou do Redis OSS. Para clusters Valkey ou Redis OSS com réplicas, recomendamos que uma memória suficiente seja disponibilizada para o Valkey ou Redis OSS.

Tópicos

- [Aumento de escala vertical de clusters do Valkey ou Redis OSS com réplicas](#)
- [Redução de escala vertical de clusters do Valkey ou Redis OSS com réplicas](#)
- [Aumento da capacidade de leitura](#)
- [Redução da capacidade de leitura](#)

Related Topics

- [Alta disponibilidade com o uso de grupos de replicação](#)
- [Replicação: Modo de cluster Valkey e Redis OSS desativado versus ativado](#)
- [Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#)
- [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#)

Tópicos

- [Aumento de escala vertical de clusters do Valkey ou Redis OSS com réplicas](#)
- [Redução de escala vertical de clusters do Valkey ou Redis OSS com réplicas](#)
- [Aumento da capacidade de leitura](#)
- [Redução da capacidade de leitura](#)

Aumento de escala vertical de clusters do Valkey ou Redis OSS com réplicas

ElastiCache A Amazon fornece suporte de console, CLI e API para escalar seu grupo de replicação Valkey ou Redis OSS (modo de cluster desativado).

Quando o processo de aumento de escala é iniciado, ElastiCache faça o seguinte:

1. Executa um grupo de replicação usando o novo tipo de nó.
2. Copia todos os dados do nó primário atual para o novo nó primário.
3. Sincroniza as novas réplicas de leitura com o novo nó primário.
4. Atualiza as entradas do DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Valkey 7.2 e superiores ou Redis OSS 5.0.5 e superiores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atendendo a solicitações recebidas. No Redis OSS versão 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.
5. Exclui os nós antigos (CLI/API: grupo de replicação). Você perceberá uma breve interrupção (alguns segundos) nas leituras e gravações dos nós antigos porque as conexões aos nós antigos serão desconectadas.

O tempo necessário para esse processo depende do tipo de nó e da quantidade de dados no seu cluster.

Conforme mostrado na tabela a seguir, sua operação de expansão do Valkey ou Redis OSS será bloqueada se você tiver uma atualização do mecanismo programada para a próxima janela de manutenção do cluster.

Operações bloqueadas do Valkey ou do Redis OSS

Operações pendentes	Operações bloqueadas
Amplie a sua capacidade	Atualização imediata do mecanismo
Atualização do mecanismo	Expansão imediata
Expansão e atualização do mecanismo	Expansão imediata
	Atualização imediata do mecanismo

Se você tiver uma operação pendente que está bloqueando você, será possível tomar uma destas providências.

- Planeje sua operação de expansão do Valkey ou Redis OSS para a próxima janela de manutenção, desmarcando a caixa de seleção Apply immediately (CLI use: `--no-apply-immediately`, API use: `ApplyImmediately=false`).
- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de expansão do Valkey ou Redis OSS.
- Adicione a atualização do mecanismo Valkey ou Redis OSS a essa modificação de cluster com a caixa de seleção Apply Immediately (Aplicar imediatamente) marcada (CLI use: `--apply-immediately`, API use: `ApplyImmediately=true`). Isso desbloqueia sua operação de expansão, fazendo com que a atualização do mecanismo seja realizada imediatamente.

As seções a seguir descrevem como escalar seu cluster OSS Valkey ou Redis com réplicas usando o ElastiCache console, o e a API AWS CLI. ElastiCache

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Valkey ou Redis OSS, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster.

Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso.

Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

Aumento de escala vertical de um cluster do Valkey ou Redis OSS com réplicas (console)

O tempo necessário para expandir para um tipo de nó maior varia dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

O processo a seguir escala seu cluster com réplicas do tipo de nó atual para um novo tipo de nó maior usando o ElastiCache console. Durante esse processo, poderá haver uma breve interrupção de leituras e gravações de outras versões do nó primário enquanto a entrada de DNS é atualizada.

Talvez você observe menos de 1 segundo de tempo de inatividade para os nós executados nas versões 5.0.5 e posteriores e alguns segundos para as versões mais antigas.

Para aumentar a escala vertical de um cluster do Valkey ou Redis OSS com réplicas (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS
3. Na lista de clusters, escolha o cluster que você deseja expandir. Esse cluster deve estar executando o mecanismo Valkey ou Redis OSS, e não o mecanismo clustered Valkey ou Redis OSS.
4. Escolha Modificar.
5. No assistente Modify Cluster:
 - a. Na lista Node type, escolha o tipo de nó a partir do qual você deseja escalar. Observe que nem todos os tipos de nó estão disponíveis para redução.
 - b. Se você estiver usando `reserved-memory` para gerenciar sua memória, na lista Parameter Group, escolha o parameter group personalizado que reserva a quantidade correta de memória para o seu novo tipo de nó.
6. Se quiser executar o processo de expansão imediatamente, escolha a caixa de seleção Apply immediately. Se a caixa de seleção Apply immediately não for escolhida, o processo de expansão será realizado durante a próxima janela de manutenção desse cluster.
7. Escolha Modificar.
8. Quando o status do cluster muda de `modifying` para `available`, significa que o seu cluster foi escalonado para o novo tipo de nó. Não é necessário atualizar os endpoints no seu aplicativo.

Aumento de escala vertical de um grupo de replicação do Valkey ou Redis OSS (AWS CLI)

O processo a seguir dimensiona o grupo de replicação do tipo de nó atual para um novo tipo de nó maior usando a AWS CLI. Durante esse processo, ElastiCache atualiza as entradas de DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Valkey 7.2 e superiores ou Redis OSS 5.0.5 e superiores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atendendo a solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.

O tempo necessário para aumentar a escala verticalmente até um tipo de nó maior varia, dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

Para aumentar a escala verticalmente de um grupo de replicação Valkey ou Redis OSS (AWS CLI)

1. Determine para quais tipos de nós você pode escalar executando o AWS CLI `list-allowed-node-type-modifications` comando com o parâmetro a seguir.
 - `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-repl-group
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-repl-group
```

O resultado dessa operação é semelhante a este (formato JSON).

```
{
  "ScaleUpModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ]
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando o AWS CLI `modify-replication-group` comando com os parâmetros a seguir.

- `--replication-group-id` - o nome do grupo de replicação.
- `--cache-node-type`: o novo tipo de nó maior dos clusters nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pelo comando `list-allowed-node-type-modifications` na etapa anterior.
- `--cache-parameter-group-name` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
- `--apply-immediately`: faz com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar a operação de expansão para a próxima janela de manutenção, use `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-repl-group \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-repl-group ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

O resultado desse comando é semelhante a este (formato JSON).

```
{
  "ReplicationGroup": {
    "Status": "available",
    "Description": "Some description",
    "NodeGroups": [{
      "Status": "available",
      "NodeGroupMembers": [{
        "CurrentRole": "primary",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
        }
      },
      "CacheClusterId": "my-repl-group-001"
    ],
    {
      "CurrentRole": "replica",
      "PreferredAvailabilityZone": "us-west-2c",
      "CacheNodeId": "0001",
      "ReadEndpoint": {
        "Port": 6379,
        "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
      }
    },
    "CacheClusterId": "my-repl-group-002"
  ],
  "NodeGroupId": "0001",
  "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
  }
}],
  "ReplicationGroupId": "my-repl-group",
  "SnapshotRetentionLimit": 1,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "12:00-13:00",
  "SnapshottingClusterId": "my-repl-group-002",
  "MemberClusters": [
    "my-repl-group-001",
    "my-repl-group-002"
  ],
}
```



```
"PendingModifiedValues": {}  
}  
}
```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou o `--apply-immediately` parâmetro, monitore o status do grupo de replicação usando o `AWS CLI describe-replication-group` comando com o parâmetro a seguir. Enquanto o status ainda for `modifying` (em modificação), você poderá observar menos de 1 segundo de tempo de inatividade para os nós executados nas versões 5.0.6 e posteriores, e uma breve interrupção de leituras e gravações para as versões mais antigas do nó primário enquanto a entrada do DNS é atualizada.

 - `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache describe-replication-groups \  
    --replication-group-id my-replication-group
```

Para Windows:

```
aws elasticache describe-replication-groups ^  
    --replication-group-id my-replication-group
```

Para obter mais informações, consulte [describe-replication-groups](#) na AWS CLI Referência.

Escalando um grupo de replicação (API) Valkey ou Redis OSS ElastiCache

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó maior usando a ElastiCache API. Para o Valkey 7.2 e superiores ou Redis OSS 5.0.5 e superiores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atendendo a solicitações recebidas. No Redis OSS versão 4.0.10 e anteriores, você pode notar uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada do DNS é atualizada.

O tempo necessário para aumentar a escala verticalmente até um tipo de nó maior varia, dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

Para escalar um grupo de replicação (API) Valkey ou Redis OSS ElastiCache

1. Determine para quais tipos de nós você pode escalar usando a `ListAllowedNodeTypeModifications` ação da ElastiCache API com o parâmetro a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação. Use esse parâmetro para descrever um determinado grupo de replicação em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ListAllowedNodeTypeModifications  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) a Amazon ElastiCache API Reference.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando a ação da `ModifyReplicationGroup` ElastiCache API e com os parâmetros a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação.
 - `CacheNodeType`: o novo tipo de nó maior dos clusters nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela ação `ListAllowedNodeTypeModifications` na etapa anterior.
 - `CacheParameterGroupName` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
 - `ApplyImmediately`: defina como `true` para fazer com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção, use `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup
```

```
&ApplyImmediately=true
&CacheNodeType=cache.m3.2xlarge
&CacheParameterGroupName=redis32-m3-2x1
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [ModifyReplicationGroup](#) a Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a `DescribeReplicationGroups` ação da ElastiCache API com os parâmetros a seguir. Quando o status mudar de `modifying` para `available`, você poderá começar a gravar no seu novo grupo de replicação expandido.

 - `ReplicationGroupId` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeReplicationGroups](#) a Amazon ElastiCache API Reference.

Redução de escala vertical de clusters do Valkey ou Redis OSS com réplicas

As seções a seguir demonstram passo a passo o processo de reduzir a escala vertical de um cluster Valkey ou Redis OSS (modo cluster desabilitado) com nós de réplica para um tipo de nó menor. Garantir que o novo tipo de nó menor seja suficientemente grande para acomodar todos os dados e a sobrecarga é muito importante para o sucesso. Para obter mais informações, consulte [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#).

Note

Para clusters que executam o tipo de nó `r6gd`, você só pode escalar para tamanhos de nó dentro da família de nós `r6gd`.

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Valkey ou Redis OSS, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster.

Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso.

Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

Redução de escala vertical de um cluster do Valkey ou Redis OSS com réplicas (console)

O processo a seguir dimensiona seu cluster OSS Valkey ou Redis com nós de réplica para um tipo de nó menor usando o console. ElastiCache

Para reduzir a escala vertical um cluster do Valkey ou Redis OSS com nós de réplica (console)

1. Certifique-se de que o tipo de nó menor seja adequado para suas necessidades de dados e sobrecarga.
2. Se o seu parameter group usar `reserved-memory` para reservar memória para a sobrecarga do Valkey ou Redis OSS, certifique-se de ter um parameter group personalizado para reservar a quantidade correta de memória para o seu novo tipo de nó.

Alternativamente, você pode modificar seu parameter group personalizado para usar `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

3. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
4. Na lista de clusters, escolha o cluster que você deseja reduzir. Esse cluster deve estar executando o mecanismo Valkey ou Redis OSS, e não o mecanismo clustered Valkey ou Redis OSS.
5. Escolha Modificar.
6. No assistente Modify Cluster:
 - a. Na lista Node type (Tipo de nó), escolha o tipo de nó para o qual deseja reduzir.
 - b. Se você estiver usando `reserved-memory` para gerenciar sua memória, na lista Parameter Group, escolha o parameter group personalizado que reserva a quantidade correta de memória para o seu novo tipo de nó.
7. Se desejar executar o processo de redução imediatamente, escolha a caixa de seleção Apply immediately (Aplicar imediatamente). Se a caixa de seleção Apply immediately (Aplicar imediatamente) não for escolhida, o processo de redução será realizado durante a próxima janela de manutenção desse cluster.
8. Escolha Modificar.
9. Quando o status do cluster muda de `modifying` para `available`, significa que o seu cluster foi escalonado para o novo tipo de nó. Não é necessário atualizar os endpoints no seu aplicativo.

Redução de escala vertical de um grupo de replicação do Valkey ou Redis OSS (AWS CLI)

O processo a seguir dimensiona grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando a AWS CLI. Durante esse processo, ElastiCache atualiza as entradas de DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Valkey 7.2 e superiores ou Redis OSS 5.0.5 e superiores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atendendo a solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.

No entanto, as leituras dos clusters de réplica de leitura continuam ininterruptas.

O tempo necessário para reduzir a escala verticalmente até um tipo de nó menor varia, dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

Para reduzir a escala verticalmente de um grupo de replicação do Valkey ou Redis OSS (AWS CLI)

1. Determine para quais tipos de nós você pode reduzir executando o AWS CLI `list-allowed-node-type-modifications` comando com o parâmetro a seguir.
 - `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-repl-group
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-repl-group
```

O resultado dessa operação é semelhante a este (formato JSON).

```
{
  "ScaleDownModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ]
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando o AWS CLI `modify-replication-group` comando com os parâmetros a seguir.
 - `--replication-group-id` - o nome do grupo de replicação.
 - `--cache-node-type`: o novo tipo de nó menor dos clusters nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pelo comando `list-allowed-node-type-modifications` na etapa anterior.
 - `--cache-parameter-group-name` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
 - `--apply-immediately`: faz com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar a operação de expansão para a próxima janela de manutenção, use `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-repl-group \  
  --cache-node-type cache.t2.small \  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-repl-group ^  
  --cache-node-type cache.t2.small ^  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

O resultado desse comando é semelhante a este (formato JSON).

```

{"ReplicationGroup": {
  "Status": "available",
  "Description": "Some description",
  "NodeGroups": [
    {
      "Status": "available",
      "NodeGroupMembers": [
        {
          "CurrentRole": "primary",
          "PreferredAvailabilityZone": "us-west-2b",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Port": 6379,
            "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
          },
          "CacheClusterId": "my-repl-group-001"
        },
        {
          "CurrentRole": "replica",
          "PreferredAvailabilityZone": "us-west-2c",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Port": 6379,
            "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
          },
          "CacheClusterId": "my-repl-group-002"
        }
      ],
      "NodeGroupId": "0001",
      "PrimaryEndpoint": {
        "Port": 6379,
        "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
      }
    }
  ],
  "ReplicationGroupId": "my-repl-group",
  "SnapshotRetentionLimit": 1,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "12:00-13:00",

```



```
    "SnapshottingClusterId": "my-repl-group-002",
    "MemberClusters": [
        "my-repl-group-001",
        "my-repl-group-002",
    ],
    "PendingModifiedValues": {}
}
```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou o `--apply-immediately` parâmetro, monitore o status do grupo de replicação usando o `AWS CLI describe-replication-group` comando com o parâmetro a seguir. Quando o status mudar de `modifying` (modificando) para `available` (disponível), você poderá começar a gravar no seu novo grupo de replicação reduzido.
- `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache describe-replication-group \
    --replication-group-id my-replication-group
```

Para Windows:

```
aws elasticache describe-replication-groups ^
    --replication-group-id my-replication-group
```

Para obter mais informações, consulte [describe-replication-groups](#) na AWS CLI Referência.

Reduzindo um grupo de replicação (API) do Valkey ou do Redis OSS ElastiCache

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando a ElastiCache API. Durante esse processo, ElastiCache atualiza as entradas de DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Valkey 7.2 e superiores ou Redis OSS 5.0.5 e superiores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atendendo a solicitações recebidas. No Redis OSS versão 4.0.10 e anteriores, você perceberá uma breve

interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada. No entanto, as leituras dos clusters de réplica de leitura continuam ininterruptas.

O tempo necessário para reduzir a escala verticalmente até um tipo de nó menor varia, dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

Para reduzir um grupo de replicação (API) Valkey ou Redis OSS ElastiCache

1. Determine quais tipos de nós você pode reduzir usando a `ListAllowedNodeTypeModifications` ação da ElastiCache API com o parâmetro a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação. Use esse parâmetro para descrever um determinado grupo de replicação em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) a Amazon ElastiCache API Reference.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando a ação da `ModifyReplicationGroup` ElastiCache API e com os parâmetros a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação.
 - `CacheNodeType`: o novo tipo de nó menor dos clusters nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela ação `ListAllowedNodeTypeModifications` na etapa anterior.
 - `CacheParameterGroupName` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.

- **ApplyImmediately**: defina como `true` para fazer com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar o processo de redução até a próxima janela de manutenção, use `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ApplyImmediately=true  
  &CacheNodeType=cache.m3.2xlarge  
  &CacheParameterGroupName=redis32-m3-2x1  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [ModifyReplicationGroup](#) a Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a `DescribeReplicationGroups` ação da ElastiCache API com os parâmetros a seguir. Quando o status mudar de `modifying` (modificando) para `available` (disponível), você poderá começar a gravar no seu novo grupo de replicação reduzido.

- **ReplicationGroupId** - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeReplicationGroups](#) a Amazon ElastiCache API Reference.

Aumento da capacidade de leitura

Para aumentar a capacidade de leitura, adicione réplicas de leitura (até um máximo de cinco) ao seu grupo de replicação Valkey ou Redis OSS.

Você pode escalar a capacidade de leitura do cluster Valkey ou Redis OSS usando o ElastiCache console AWS CLI, o ou a API. ElastiCache Para obter mais informações, consulte [Adição de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#).

Redução da capacidade de leitura

Para diminuir a capacidade de leitura, exclua uma ou mais réplicas de leitura do seu cluster Valkey ou Redis OSS com réplicas (chamado de grupo de replicação na API/CLI). Se o cluster for habilitado para Multi-AZ com failover automático, não será possível excluir a última réplica de leitura sem primeiro desabilitar o Multi-AZ. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

Para obter mais informações, consulte [Exclusão de uma réplica de leitura do Valkey ou do Redis OSS \(modo cluster desabilitado\)](#).

Escalabilidade de clusters no Valkey ou Redis OSS (modo cluster habilitado)

À medida que a demanda nos clusters muda, você pode optar por aumentar a performance ou reduzir os custos alterando o número de fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado). Recomendamos o uso da escalabilidade horizontal online para esse ajuste, pois permite que o seu cluster continue a atender às solicitações durante o processo de escalabilidade.

As condições sob as quais você pode decidir redimensionar seu cluster incluem o seguinte:

- Uso intenso de memória:

Se os nós no cluster estão sob uso intenso da memória, você pode optar por aumentar a escala e ter mais recursos para melhor armazenar dados e atender a solicitações.

Você pode determinar se seus nós estão sob pressão de memória monitorando as seguintes métricas: `FreeableMemorySwapUsage`, `BytesUsedForCache`.

- CPU ou gargalo de rede:

Se os latency/throughput problemas estiverem afetando seu cluster, talvez seja necessário escalar para resolvê-los.

Você pode monitorar seus níveis de latência e taxa de transferência monitorando as seguintes métricas: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOutCurrConnections`, e `NewConnections`.

- Seu cluster está acima da escala:

A demanda atual no cluster permite que haja uma redução na escala sem afetar o desempenho e proporcionando corte de custos.

Você pode monitorar o uso do seu cluster para determinar se você pode ou não escalar com segurança usando as seguintes métricas: `FreeableMemorySwapUsage`, `BytesUsedForCache`, `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`.

Impacto da escalabilidade no desempenho

Quando você altera a escala usando o processo offline, seu cluster fica offline para uma parte significativa do processo e, por conseguinte, não é capaz de atender a solicitações. Quando você altera a escala usando o método online, como a escalabilidade é uma operação com uso intensivo de computação, há queda no desempenho, mas ainda assim seu cluster continua atendendo a

solicitações durante a operação de escalabilidade. O quanto o desempenho é afetado depende do seu uso normal da CPU e dos seus dados.

Existem duas maneiras de escalar seu cluster do Valkey ou Redis OSS (modo cluster desabilitado): com escalabilidade horizontal e vertical.

- A escalabilidade horizontal permite alterar o número de grupos de nós (estilhaços) no grupo de replicação adicionando ou removendo grupos de nós (estilhaços). O processo de refragmentação on-line permite a escalabilidade in/out enquanto o cluster continua atendendo às solicitações recebidas.

Configurar os slots em seu novo cluster de forma diferente como que eles foram configurados no cluster antigo. Apenas no método offline.

- Escalabilidade vertical — altere o tipo de nó para redimensionar o cluster. O escalonamento vertical on-line permite o escalonamento up/down enquanto o cluster continua atendendo às solicitações recebidas.

Se estiver reduzindo o tamanho e a capacidade de memória do cluster, reduzindo ou expandindo, garanta que a nova configuração tenha memória suficiente para seus dados e a sobrecarga do Valkey ou Redis OSS.

Para obter mais informações, consulte [Escolha do tamanho do nó](#).

Sumário

- [Refragmentação offline para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#)
- [Refragmentação online para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#)
 - [Adição de fragmentos com refragmentação online](#)
 - [Remoção de fragmentos com refragmentação online](#)
 - [Remoção de fragmentos \(console\)](#)
 - [Remoção de fragmentos \(AWS CLI\)](#)
 - [Removendo fragmentos \(ElastiCacheAPI\)](#)
 - [Rebalanceamento de frgamentos online](#)
 - [Rebalanceamento de fragmentos online \(Console\)](#)
 - [Rebalanceamento de fragmentos online \(AWS CLI\)](#)
 - [Rebalanceamento de fragmentos on-line \(API\) ElastiCache](#)

- [Escalabilidade vertical online com modificação do tipo de nó](#)
 - [Aumento de escala vertical online](#)
 - [Aumento de escala na vertical de clusters do Valkey ou Redis OSS \(console\)](#)
 - [Aumento da escala na vertical de clusters do Valkey ou Redis OSS \(AWS CLI\)](#)
 - [Ampliando clusters OSS Valkey ou Redis \(API\) ElastiCache](#)
 - [Redução de escala vertical online](#)
 - [Redução de escala na vertical de clusters do Valkey ou Redis OSS \(console\)](#)
 - [Redução de escala na vertical de clusters do Valkey ou Redis OSS \(AWS CLI\)](#)
 - [Reduzindo a escala de clusters OSS Valkey ou Redis \(API\) ElastiCache](#)

Refragmentação offline para o Valkey ou o Redis OSS (modo cluster habilitado)

A principal vantagem de obter a reconfiguração de estilhaços offline é que você pode fazer mais do que simplesmente adicionar ou remover estilhaços do seu grupo de replicação. Quando você faz a refragmentação e o rebalanceamento offline, além de alterar o número de fragmentos no seu grupo de replicação, é possível fazer o seguinte:

 Note

A refragmentação offline não é compatível com clusters do Valkey ou Redis OSS que tenham a classificação de dados em níveis ativada. Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

- Alterar o tipo de nó do seu grupo de replicação.
- Especificar a zona de disponibilidade para cada nó no grupo de replicação.
- Fazer o upgrade para uma versão mais recente do mecanismo.
- Especificar o número de nós de réplica em cada estilhaço de forma independente.
- Especificar o espaço de chaves para cada estilhaço.

A principal desvantagem da reconfiguração de fragmentos offline é que o cluster fica offline começando com a parte de restauração do processo e continua até você atualizar os endpoints no aplicativo. O tempo em que o cluster fica offline depende da quantidade de dados no seu cluster.

Para reconfigurar seus fragmentos offline no cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Crie um backup manual do seu cluster Valkey ou Redis OSS existente. Para obter mais informações, consulte [Realização de backups manuais](#).
2. Crie um novo cluster fazendo a restauração a partir do backup. Para obter mais informações, consulte [Restauração de um backup para um novo cache](#).
3. Atualize os endpoints no seu aplicativo para os endpoints do novo cluster. Para obter mais informações, consulte [Encontrar endpoints de conexão no ElastiCache](#).

Refragmentação online para o Valkey ou o Redis OSS (modo cluster habilitado)

Ao usar a refragmentação on-line e o rebalanceamento de fragmentos com o ElastiCache Valkey 7.2 ou mais recente, ou o Redis OSS versão 3.2.10 ou mais recente, você pode escalar dinamicamente seu cluster Valkey ou Redis OSS (modo de cluster ativado) sem tempo de inatividade. Essa abordagem significa que seu cluster pode continuar atendendo a solicitações mesmo durante a escalabilidade ou o rebalanceamento.

Você pode fazer o seguinte:

- Aumentar a escala horizontalmente: aumente a capacidade de leitura e gravação adicionando fragmentos (grupos de nós) ao seu cluster do Valkey ou Redis OSS (modo cluster habilitado) (grupo de replicação).

Se você adicionar um ou mais fragmentos ao grupo de replicação, o número de nós em cada novo fragmento será o mesmo que o número de nós no menor dos fragmentos existentes.

- Reduzir a escala horizontalmente: reduza de capacidade de leitura e gravação e, por conseguinte, os custos, removendo fragmentos do cluster do Valkey ou Redis OSS (modo cluster habilitado).
- Rebalancear: mova os espaços de chave entre os fragmentos em seu cluster do Valkey ou Redis OSS (modo cluster habilitado) a fim de que eles sejam distribuídos de modo uniforme entre os fragmentos.

Você não pode fazer o seguinte:

- Configurar os fragmentos de forma independente:

Não é possível especificar o espaço de chave para os fragmentos de forma independente. Para fazer isso, você deve usar o processo offline.

Atualmente, as seguintes limitações se aplicam à fragmentação e ao rebalanceamento ElastiCache on-line:

- Esses processos exigem o Valkey 7.2 ou mais recente ou Redis OSS 3.2.10 ou mais recente. Para obter informações sobre como atualizar sua versão de mecanismo, consulte [Gerenciamento de versões para ElastiCache](#).
- Há limitações em relação a slots ou espaços de chave e itens grandes:

Se qualquer uma das chaves em um fragmento contiver um item grande, essa chave não será migrada para um novo fragmento durante o aumento da escala ou o rebalanceamento. Essa funcionalidade pode resultar em fragmentos desbalanceados.

Se qualquer uma das chaves em um fragmento contiver um item grande (itens maiores do que 256 MB após a serialização), o fragmento não será excluído na redução da escala. Essa funcionalidade pode resultar na não exclusão de alguns fragmentos.

- Ao aumentar a escala, o número de nós em novos fragmentos fica igual ao número de nós no menor dos fragmentos existentes.
- No aumento de escala, qualquer tag comum a todos os fragmentos existentes é copiada para os novos fragmentos.
- Ao escalar um cluster de armazenamento de dados global, não ElastiCache replicará automaticamente as funções de um dos nós existentes para o (s) novo (s) nó (s). Recomendamos carregar suas funções nos novos fragmentos depois de aumentar a escala horizontalmente do seu cluster para que todos os fragmentos tenham as mesmas funções.

Note

No ElastiCache Valkey 7.2 e superior e no Redis OSS versão 7 e ElastiCache superior: ao escalar seu cluster, ElastiCache replicará automaticamente as funções carregadas em um dos nós existentes (selecionados aleatoriamente) para o (s) novo (s) nó (s). Se sua aplicação usa [Functions](#) (Funções), recomendamos carregar todas as suas funções em todos os fragmentos antes de aumentar a escala horizontalmente, para que seu cluster não tenha definições de função diferentes em fragmentos diferentes.

Para obter mais informações, consulte [Redimensionamento de cluster on-line](#).

Você pode escalar ou reequilibrar horizontalmente seus clusters Valkey ou Redis OSS (modo de cluster ativado) usando a Console de gerenciamento da AWS, a e a API.AWS CLI ElastiCache

Adição de fragmentos com refragmentação online

Você pode adicionar fragmentos ao seu cluster Valkey ou Redis OSS (modo de cluster ativado) usando a API Console de gerenciamento da AWS,AWS CLI ou. ElastiCache Quando você adiciona fragmentos a um cluster do Valkey ou Redis OSS (modo cluster habilitado), qualquer tag nos fragmentos existentes é copiada para os novos fragmentos.

Adição de fragmentos (console)

Você pode usar o Console de gerenciamento da AWS para adicionar um ou mais fragmentos ao seu cluster Valkey ou Redis OSS (modo de cluster ativado). O procedimento a seguir descreve o processo.

Como adicionar fragmentos ao seu cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Localize e escolha o nome, não a caixa à esquerda do nome do cluster, do cluster do Valkey ou Redis OSS (modo cluster habilitado) ao qual você deseja adicionar fragmentos.

Tip

O Valkey ou Redis OSS (modo cluster habilitado) exibe Clustered Valkey ou Clustered Redis OSS na coluna Mode (Modo)

4. Escolha Add shard (Adicionar fragmento).
 - a. Para Number of shards to be added, escolha o número de fragmentos que deseja adicionar a este cluster.
 - b. Para Availability zone(s) (Zonas de disponibilidade), escolha No preference (Sem preferência) ou Specify availability zones (Especificar zonas de disponibilidade).
 - c. Se você optar por Specify availability zones (Especificar zona de disponibilidade), para cada nó em cada fragmento, selecione a zona de disponibilidade do nó na lista de zonas de disponibilidade.
 - d. Escolha Add (Adicionar).

Adição de fragmentos (AWS CLI)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado) adicionando fragmentos por meio da AWS CLI.

Use os parâmetros a seguir com `modify-replication-group-shard-configuration`.

Parâmetros

- `--apply-immediately`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `--replication-group-id`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `--node-group-count`: obrigatório. Especifica o número de fragmentos (grupos de nós) que devem existir quando a operação for concluída. Ao adicionar fragmentos, o valor de `--node-group-count` deve ser maior que o número atual de fragmentos.

Como opção, você pode especificar a zona de disponibilidade para cada nó no grupo de replicação usando `--resharding-configuration`.

- `--resharding-configuration`: opcional. Uma lista de zonas de disponibilidade preferenciais para cada nó em cada fragmento no grupo de replicação. Use este parâmetro somente se o valor de `--node-group-count` for maior que o número atual de fragmentos. Se esse parâmetro for omitido ao adicionar fragmentos, a Amazon ElastiCache selecionará as zonas de disponibilidade para os novos nós.

O exemplo a seguir reconfigura os espaços de chave em quatro fragmentos em um cluster do Valkey ou Redis OSS (modo cluster habilitado) `my-cluster`. O exemplo também especifica a zona de disponibilidade para cada nó em cada fragmento. A operação começa imediatamente.

Example Adicionar fragmentos

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group-shard-configuration \
  --replication-group-id my-cluster \
  --node-group-count 4 \
  --resharding-configuration \
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" \
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" \
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" \
```

```
"PreferredAvailabilityZones=us-east-2d,us-east-2c" \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
--replication-group-id my-cluster ^  
--node-group-count 4 ^  
--resharding-configuration ^  
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^  
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^  
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^  
    "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^  
--apply-immediately
```

Para obter mais informações, consulte [modify-replication-group-shard-configuration](#) na AWS CLI documentação.

Adicionar fragmentos (ElastiCache API)

Você pode usar a ElastiCache API para reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) on-line usando a operação.

ModifyReplicationGroupShardConfiguration

Use os parâmetros a seguir com ModifyReplicationGroupShardConfiguration.

Parâmetros

- ApplyImmediately=true: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- ReplicationGroupId: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- NodeGroupCount: obrigatório. Especifica o número de fragmentos (grupos de nós) que devem existir quando a operação for concluída. Ao adicionar fragmentos, o valor de NodeGroupCount deve ser maior que o número atual de fragmentos.

Como opção, você pode especificar a zona de disponibilidade para cada nó no grupo de replicação usando ReshardingConfiguration.

- ReshardingConfiguration: opcional. Uma lista de zonas de disponibilidade preferenciais para cada nó em cada fragmento no grupo de replicação. Use este parâmetro somente se o valor de

NodeGroupCount for maior que o número atual de fragmentos. Se esse parâmetro for omitido ao adicionar fragmentos, a Amazon ElastiCache selecionará as zonas de disponibilidade para os novos nós.

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) adicionando fragmentos usando a API. ElastiCache

Example Adicionar fragmentos

O exemplo a seguir adiciona grupos de nós ao cluster do Valkey ou Redis OSS (modo cluster habilitado) `my-cluster` de modo que haja um total de quatro grupos de nós quando a operação for concluída. O exemplo também especifica a zona de disponibilidade para cada nó em cada fragmento. A operação começa imediatamente.

```
https://elasticache.us-east-2.amazonaws.com/  
  ?Action=ModifyReplicationGroupShardConfiguration  
  &ApplyImmediately=true  
  &NodeGroupCount=4  
  &ReplicationGroupId=my-cluster  
  
  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone  
east-2a  
  
  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  
  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone  
east-2b  
  
  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone  
east-2a  
  
  &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  
  &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone  
east-2d  
  
  &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone  
east-2d
```

```
&ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZoneName=
east-2c
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20171002T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyReplicationGroupShardConfiguration](#) na Referência ElastiCache da API.

Remoção de fragmentos com refragmentação online

Você pode remover fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado) usando a API Console de gerenciamento da AWS, AWS CLI ou ElastiCache.

Tópicos

- [Remoção de fragmentos \(console\)](#)
- [Remoção de fragmentos \(AWS CLI\)](#)
- [Removendo fragmentos \(ElastiCacheAPI\)](#)

Remoção de fragmentos (console)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado) removendo fragmentos por meio da Console de gerenciamento da AWS.

Antes de remover grupos de nós (fragmentos) do seu grupo de replicação, certifique-se ElastiCache de que todos os seus dados caibam nos fragmentos restantes. Se os dados couberem, os fragmentos especificados serão excluídos do grupo de replicação como solicitado. Se os dados não couberem nos grupos de nós restantes, o processo será encerrado e o grupo de replicação será deixado com a mesma configuração do grupo de nós anterior à solicitação.

Você pode usar o Console de gerenciamento da AWS para remover um ou mais fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado). Não é possível remover todos os fragmentos de um grupo de replicação. Nesse caso, você deve excluir o grupo de replicação. Para obter mais informações, consulte [Exclusão de um grupo de replicação](#). O procedimento a seguir descreve o processo para excluir um ou mais fragmentos.

Para remover fragmentos do seu cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Localize e escolha o nome, não a caixa à esquerda do nome do cluster, do cluster do Valkey ou Redis OSS (modo cluster habilitado) do qual você deseja remover fragmentos.

 Tip

Os clusters do Valkey ou Redis OSS (modo cluster habilitado) têm um valor igual ou superior a 1 na coluna Shards (Fragmentos).

4. Na lista de fragmentos, escolha a caixa à esquerda do nome de cada fragmento que você gostaria de excluir.
5. Escolha Delete shard (Excluir fragmento).

Remoção de fragmentos (AWS CLI)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado) removendo fragmentos por meio da AWS CLI.

 Important

Antes de remover grupos de nós (fragmentos) do seu grupo de replicação, certifique-se ElastiCache de que todos os seus dados caibam nos fragmentos restantes. Se os dados couberem, os fragmentos especificados (`--node-groups-to-remove`) serão excluídos do grupo de replicação como solicitado e seus espaços de chave serão mapeados para os fragmentos restantes. Se os dados não couberem nos grupos de nós restantes, o processo será encerrado e o grupo de replicação será deixado com a mesma configuração do grupo de nós anterior à solicitação.

Você pode usar o AWS CLI para remover um ou mais fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado). Não é possível remover todos os fragmentos de um grupo de replicação. Nesse caso, você deve excluir o grupo de replicação. Para obter mais informações, consulte [Exclusão de um grupo de replicação](#).

Use os parâmetros a seguir com `modify-replication-group-shard-configuration`.

Parâmetros

- `--apply-immediately`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `--replication-group-id`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `--node-group-count`: obrigatório. Especifica o número de fragmentos (grupos de nós) que devem existir quando a operação for concluída. Ao remover fragmentos, o valor de `--node-group-count` deve ser menor que o número atual de fragmentos.
- `--node-groups-to-remove`: obrigatório quando `--node-group-count` for menor do que o número atual de grupos de nós (fragmentos). Uma lista de fragmentos (grupo de nós) IDs a serem removidos do grupo de replicação.

O procedimento a seguir descreve o processo para excluir um ou mais fragmentos.

Example Remover fragmentos

O exemplo a seguir remove dois grupos de nós do cluster do Valkey ou Redis OSS (modo cluster habilitado) `my-cluster` de modo que haja um total de dois grupos de nós quando a operação for concluída. Os espaços de chave dos fragmentos removidos são distribuídos uniformemente entre os fragmentos restantes.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group-shard-configuration \
  --replication-group-id my-cluster \
  --node-group-count 2 \
  --node-groups-to-remove "0002" "0003" \
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
  --replication-group-id my-cluster ^
  --node-group-count 2 ^
  --node-groups-to-remove "0002" "0003" ^
  --apply-immediately
```

Removendo fragmentos (ElastiCacheAPI)

Você pode usar a ElastiCache API para reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) on-line usando a operação.

ModifyReplicationGroupShardConfiguration

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) removendo fragmentos usando a API. ElastiCache

Important

Antes de remover grupos de nós (fragmentos) do seu grupo de replicação, certifique-se ElastiCache de que todos os seus dados caibam nos fragmentos restantes. Se os dados couberem, os fragmentos especificados (NodeGroupsToRemove) serão excluídos do grupo de replicação como solicitado e seus espaços de chave serão mapeados para os fragmentos restantes. Se os dados não couberem nos grupos de nós restantes, o processo será encerrado e o grupo de replicação será deixado com a mesma configuração do grupo de nós anterior à solicitação.

Você pode usar a ElastiCache API para remover um ou mais fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado). Não é possível remover todos os fragmentos de um grupo de replicação. Nesse caso, você deve excluir o grupo de replicação. Para obter mais informações, consulte [Exclusão de um grupo de replicação](#).

Use os parâmetros a seguir com ModifyReplicationGroupShardConfiguration.

Parâmetros

- **ApplyImmediately=true:** obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- **ReplicationGroupId:** obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- **NodeGroupCount:** obrigatório. Especifica o número de fragmentos (grupos de nós) que devem existir quando a operação for concluída. Ao remover fragmentos, o valor de NodeGroupCount deve ser menor que o número atual de fragmentos.
- **NodeGroupsToRemove:** obrigatório quando --node-group-count for menor do que o número atual de grupos de nós (fragmentos). Uma lista de fragmentos (grupo de nós) IDs a serem removidos do grupo de replicação.

O procedimento a seguir descreve o processo para excluir um ou mais fragmentos.

Example Remove fragmentos

O exemplo a seguir remove dois grupos de nós do cluster do Valkey ou Redis OSS (modo cluster habilitado) `my-cluster` de modo que haja um total de dois grupos de nós quando a operação for concluída. Os espaços de chave dos fragmentos removidos são distribuídos uniformemente entre os fragmentos restantes.

```
https://elasticache.us-east-2.amazonaws.com/  
?Action=ModifyReplicationGroupShardConfiguration  
&ApplyImmediately=true  
&NodeGroupCount=2  
&ReplicationGroupId=my-cluster  
&NodeGroupsToRemove.member.1=0002  
&NodeGroupsToRemove.member.2=0003  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20171002T192317Z  
&X-Amz-Credential=<credential>
```

Rebalanceamento de fragmentos online

Você pode reequilibrar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) usando a Console de gerenciamento da AWS API, ou AWS CLI ElastiCache

Tópicos

- [Rebalanceamento de fragmentos online \(Console\)](#)
- [Rebalanceamento de fragmentos online \(AWS CLI\)](#)
- [Rebalanceamento de fragmentos on-line \(API\) ElastiCache](#)

Rebalanceamento de fragmentos online (Console)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado) rebalanceando fragmentos por meio da Console de gerenciamento da AWS.

Para rebalancear os espaços de chave entre os fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Escolha o nome do cluster (não a caixa à esquerda do nome do cluster) do Valkey ou Redis OSS (modo cluster habilitado) que você gostaria de rebalancear.

 Tip

Os clusters do Valkey ou Redis OSS (modo cluster habilitado) têm um valor igual ou superior a 1 na coluna Shards (Fragmentos).

4. Escolha Rebalance (Rebalancear).
5. Quando solicitado, escolha Rebalance (Rebalancear). Você pode ver uma mensagem semelhante a esta: *Slots in the replication group are uniformly distributed. Nothing to do. (Service: AmazonElastiCache; Status Code: 400; Error Code: InvalidReplicationGroupState; Request ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*. Se isso acontecer, escolha Cancel (Cancelar).

Rebalanceamento de fragmentos online (AWS CLI)

Use os parâmetros a seguir com `modify-replication-group-shard-configuration`.

Parâmetros

- `-apply-immediately`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `--replication-group-id`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `--node-group-count`: obrigatório. Para rebalancear os espaços de chave entre todos os fragmentos no cluster, esse valor deve ser igual ao número atual de fragmentos.

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Valkey ou Redis OSS (modo cluster habilitado) rebalanceando fragmentos por meio da AWS CLI.

Example- Rebalanceando os fragmentos em um cluster

O exemplo a seguir rebalanceia os slots no cluster do Valkey ou Redis OSS (modo cluster habilitado) `my-cluster` de modo que os slots sejam distribuídos da forma mais uniforme possível. O valor de `--node-group-count` (4) é o número de fragmentos atualmente no cluster.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
  --replication-group-id my-cluster ^  
  --node-group-count 4 ^  
  --apply-immediately
```

Rebalanceamento de fragmentos on-line (API) ElastiCache

Você pode usar a ElastiCache API para reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) on-line usando a operação.

ModifyReplicationGroupShardConfiguration

Use os parâmetros a seguir com `ModifyReplicationGroupShardConfiguration`.

Parâmetros

- `ApplyImmediately=true`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `ReplicationGroupId`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `NodeGroupCount`: obrigatório. Para rebalancear os espaços de chave entre todos os fragmentos no cluster, esse valor deve ser igual ao número atual de fragmentos.

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) reequilibrando os fragmentos usando a API. ElastiCache

Example- Rebalanceamento do cluster

O exemplo a seguir rebalanceia os slots no cluster do Valkey ou Redis OSS (modo cluster habilitado) `my-cluster` de modo que os slots sejam distribuídos da forma mais uniforme possível. O valor de `NodeGroupCount` (4) é o número de fragmentos atualmente no cluster.

```
https://elasticache.us-east-2.amazonaws.com/  
?Action=ModifyReplicationGroupShardConfiguration  
&ApplyImmediately=true  
&NodeGroupCount=4  
&ReplicationGroupId=my-cluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20171002T192317Z  
&X-Amz-Credential=<credential>
```

Escalabilidade vertical online com modificação do tipo de nó

Ao usar a escalabilidade vertical on-line com o Valkey versão 7.2 ou mais recente, ou Redis OSS versão 3.2.10 ou mais recente, você pode escalar clusters do Valkey ou Redis OSS dinamicamente com tempo de inatividade mínimo. Isso permite que o cluster do Valkey ou Redis OSS veicule solicitações mesmo ao ser escalado.

Note

Não há compatibilidade com escalabilidade entre um cluster de classificação de dados em níveis (p. ex., um cluster que use um tipo de nó `r6gd`) e um cluster sem classificação de dados em níveis (p. ex., um cluster que use um tipo de nó `r6g`). Para obter mais informações, consulte [Classificação de dados em níveis no ElastiCache](#).

Você pode fazer o seguinte:

- Aumentar a escala verticalmente: aumente a capacidade de leitura e gravação ajustando o tipo de nó do cluster do Valkey ou Redis OSS para usar um tipo de nó maior.

ElastiCache redimensiona dinamicamente seu cluster enquanto permanece on-line e atende às solicitações.

- Redução de escala vertical: reduza a capacidade de leitura e gravação ajustando o tipo de nó para usar um nó menor. Novamente, redimensiona ElastiCache dinamicamente seu cluster enquanto permanece on-line e atende às solicitações. Nesse caso, você reduz os custos diminuindo o nó.

Note

Os processos de expansão e redução dependem da criação de clusters com tipos de nó recém-selecionados e da sincronização dos novos nós com os anteriores. Para garantir um up/down fluxo de escala suave, faça o seguinte:

- Verifique se você tem capacidade suficiente de ENI (interface de rede elástica). Se estiver reduzindo, garanta que o nó menor tenha memória suficiente para absorver o tráfego esperado.

Para ver as melhores práticas para o gerenciamento da memória, consulte [Gerenciamento de memória reservada para Valkey e Redis OSS](#).

- Embora o processo de escalabilidade vertical seja desenvolvido para permanecer totalmente online, ele depende da sincronização dos dados entre o nó antigo e o novo nó. Recomendamos iniciar a expansão/redução no horário em que você acredita que o tráfego de dados seja mínimo.
- Teste o comportamento de seu aplicativo durante a escalabilidade em um ambiente de preparação, se possível.

Sumário

- [Aumento de escala vertical online](#)
 - [Aumento de escala na vertical de clusters do Valkey ou Redis OSS \(console\)](#)
 - [Aumento da escala na vertical de clusters do Valkey ou Redis OSS \(AWS CLI\)](#)
 - [Ampliando clusters OSS Valkey ou Redis \(API\) ElastiCache](#)
- [Redução de escala vertical online](#)
 - [Redução de escala na vertical de clusters do Valkey ou Redis OSS \(console\)](#)
 - [Redução de escala na vertical de clusters do Valkey ou Redis OSS \(AWS CLI\)](#)
 - [Reduzindo a escala de clusters OSS Valkey ou Redis \(API\) ElastiCache](#)

Aumento de escala vertical online

Tópicos

- [Aumento de escala na vertical de clusters do Valkey ou Redis OSS \(console\)](#)
- [Aumento da escala na vertical de clusters do Valkey ou Redis OSS \(AWS CLI\)](#)
- [Ampliando clusters OSS Valkey ou Redis \(API\) ElastiCache](#)

Aumento de escala na vertical de clusters do Valkey ou Redis OSS (console)

O procedimento a seguir descreve como escalar um cluster OSS Valkey ou Redis usando o ElastiCache Management Console. Durante esse processo, o cluster continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala verticalmente de um cluster do Valkey ou Redis OSS (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Na lista de clusters, escolha o cluster.
4. Escolha Modificar.
5. No assistente Modify Cluster:
 - Na lista Node type, escolha o tipo de nó a partir do qual você deseja escalar. Para expandir, selecione um tipo de nó maior do que o nó existente.
6. Se quiser executar o processo de expansão imediatamente, escolha a caixa Apply immediately (Aplicar imediatamente). Se a caixa Apply immediately não for escolhida, o processo de expansão será realizado durante a próxima janela de manutenção desse cluster.
7. Escolha Modificar.

Se você escolheu Apply immediately na etapa anterior, o status do cluster mudará para modifying. Quando o status mudar para available, a modificação estará completa, e você poderá começar a usar o novo cluster.

Aumento da escala na vertical de clusters do Valkey ou Redis OSS (AWS CLI)

O procedimento a seguir descreve como aumentar a escala verticalmente de um cluster do Valkey ou Redis OSS usando a AWS CLI. Durante esse processo, o cluster continuará a atender solicitações com tempo de inatividade mínimo.

Como aumentar a escala verticalmente de um cluster do Valkey ou Redis OSS (AWS CLI)

1. Determine os tipos de nós para os quais você pode escalar executando o AWS CLI `list-allowed-node-type-modifications` comando com o parâmetro a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-replication-group-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^
  --replication-group-id my-replication-group-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{
  "ScaleUpModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ],
  "ScaleDownModifications": [
    "cache.t2.micro",
```

```
        "cache.t2.small ",
        "cache.t2.medium",
        "cache.t1.small "
    ],
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu grupo de replicação para escalar até o novo tipo de nó maior usando o AWS CLI `modify-replication-group` comando e os parâmetros a seguir.
 - `--replication-group-id`: o nome do grupo de replicação que você está aumentando a escala vertical.
 - `--cache-node-type`: o novo tipo de nó para o qual você deseja escalar o cluster. Esse valor deve ser um dos tipos de nós retornados pelo comando `list-allowed-node-type-modifications` na etapa 1.
 - `--cache-parameter-group-name` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
 - `--apply-immediately`: faz com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção do cluster, use o parâmetro `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \
    --replication-group-id my-redis-cluster \
    --cache-node-type cache.m3.xlarge \
    --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
    --replication-group-id my-redis-cluster ^
    --cache-node-type cache.m3.xlarge ^
```

```
--apply-immediately
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{
  "ReplicationGroup": {
    "Status": "modifying",
    "Description": "my-redis-cluster",
    "NodeGroups": [
      {
        "Status": "modifying",
        "Slots": "0-16383",
        "NodeGroupId": "0001",
        "NodeGroupMembers": [
          {
            "PreferredAvailabilityZone": "us-east-1f",
            "CacheNodeId": "0001",
            "CacheClusterId": "my-redis-cluster-0001-001"
          },
          {
            "PreferredAvailabilityZone": "us-east-1d",
            "CacheNodeId": "0001",
            "CacheClusterId": "my-redis-cluster-0001-002"
          }
        ]
      }
    ],
    "ConfigurationEndpoint": {
      "Port": 6379,
      "Address": "my-redis-cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "my-redis-cluster",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "07:30-08:30",
    "MemberClusters": [
      "my-redis-cluster-0001-001",
      "my-redis-cluster-0001-002"
    ],
    "CacheNodeType": "cache.m3.xlarge",
  }
}
```

```
    "DataTiering": "disabled"  
    "PendingModifiedValues": {}  
  }  
}
```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou o `--apply-immediately`, verifique o status do cluster usando o AWS CLI `describe-cache-clusters` comando com o parâmetro a seguir. Quando o status mudar para `available`, você poderá começar a usar o novo nó de cluster maior.

Ampliando clusters OSS Valkey ou Redis (API) ElastiCache

O processo a seguir escala seu cluster do tipo de nó atual para um novo tipo de nó maior usando a ElastiCache API. Durante esse processo, ElastiCache atualiza as entradas de DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Valkey 7.2 e superiores, Redis OSS 5.0.5 e superiores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atendendo a solicitações recebidas. No Redis OSS versão 4.0.10 e anteriores, você pode notar uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada do DNS é atualizada.

O tempo necessário para aumentar a escala verticalmente até um tipo de nó maior varia, dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

Para escalar um cluster de cache (API) Valkey ou Redis OSS ElastiCache

1. Determine para quais tipos de nós você pode escalar usando a `ListAllowedNodeTypeModifications` ação da ElastiCache API com o parâmetro a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação. Use esse parâmetro para descrever um determinado grupo de replicação em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) a Amazon ElastiCache API Reference.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando a ação da `ModifyReplicationGroup` ElastiCache API e com os parâmetros a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação.
 - `CacheNodeType`: o novo tipo de nó maior dos clusters nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela ação `ListAllowedNodeTypeModifications` na etapa anterior.
 - `CacheParameterGroupName` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
 - `ApplyImmediately`: defina como `true` para fazer com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção, use `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ApplyImmediately=true  
&CacheNodeType=cache.m3.2xlarge  
&CacheParameterGroupName=redis32-m3-2x1  
&ReplicationGroupId=myReplGroup  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&Version=2014-12-01  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [ModifyReplicationGroup](#) a Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a `DescribeReplicationGroups` ação da ElastiCache API com os parâmetros a seguir. Quando o status mudar de `modifying` para `available`, você poderá começar a gravar no seu novo grupo de replicação expandido.
- `ReplicationGroupId` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeReplicationGroups](#) a Amazon ElastiCache API Reference.

Redução de escala vertical online

Tópicos

- [Redução de escala na vertical de clusters do Valkey ou Redis OSS \(console\)](#)
- [Redução de escala na vertical de clusters do Valkey ou Redis OSS \(AWS CLI\)](#)
- [Reduzindo a escala de clusters OSS Valkey ou Redis \(API\) ElastiCache](#)

Redução de escala na vertical de clusters do Valkey ou Redis OSS (console)

O procedimento a seguir descreve como reduzir um cluster OSS Valkey ou Redis usando o ElastiCache Management Console. Durante esse processo, o cluster do Valkey ou Redis OSS continuará a atender solicitações com tempo de inatividade mínimo.

Para reduzir a escala verticalmente de um cluster do Valkey ou Redis OSS (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Na lista de clusters, escolha seu cluster preferido.
4. Escolha Modificar.
5. No assistente Modify Cluster:
 - Na lista Node type, escolha o tipo de nó a partir do qual você deseja escalar. Para reduzir, selecione um tipo de nó menor do que o nó existente. Observe que nem todos os tipos de nó estão disponíveis para redução.
6. Se quiser executar o processo de redução imediatamente, escolha a caixa Apply immediately (Aplicar imediatamente). Se a caixa Apply immediately (Aplicar imediatamente) não for escolhida, o processo de redução será realizado durante a próxima janela de manutenção desse cluster.
7. Escolha Modificar.

Se você escolheu Apply immediately na etapa anterior, o status do cluster mudará para modifying. Quando o status mudar para available, a modificação estará completa, e você poderá começar a usar o novo cluster.

Redução de escala na vertical de clusters do Valkey ou Redis OSS (AWS CLI)

O procedimento a seguir descreve como reduzir a escala verticalmente de um cluster do Valkey ou Redis OSS usando a AWS CLI. Durante esse processo, o cluster continuará a atender solicitações com tempo de inatividade mínimo.

Como reduzir a escala verticalmente de um cluster do Valkey ou Redis OSS (AWS CLI)

1. Determine os tipos de nós para os quais você pode reduzir executando o AWS CLI `list-allowed-node-type-modifications` comando com o parâmetro a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \
  --replication-group-id my-replication-group-id
```


Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-replication-group-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ],  
  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small"  
  ]  
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu grupo de replicação para reduzi-lo ao novo tipo de nó menor, usando o AWS CLI `modify-replication-group` comando e os parâmetros a seguir.
 - `--replication-group-id`: o nome do grupo de replicação que você está reduzindo a escala na vertical.

- `--cache-node-type`: o novo tipo de nó para o qual você deseja escalar o cluster. Esse valor deve ser um dos tipos de nós retornados pelo comando `list-allowed-node-type-modifications` na etapa 1.
- `--cache-parameter-group-name` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.
- `--apply-immediately`: faz com que o processo de aumento de escala vertical seja aplicado imediatamente. Para adiar o processo de redução para a próxima janela de manutenção do cluster, use o parâmetro `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-cluster \  
  --cache-node-type cache.t2.micro \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-redis-cluster ^  
  --cache-node-type cache.t2.micro ^  
  --apply-immediately
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "my-redis-cluster",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "Slots": "0-16383",  
        "NodeGroupId": "0001",
```

```

        "NodeGroupMembers": [
            {
                "PreferredAvailabilityZone": "us-east-1f",
                "CacheNodeId": "0001",
                "CacheClusterId": "my-redis-cluster-0001-001"
            },
            {
                "PreferredAvailabilityZone": "us-east-1d",
                "CacheNodeId": "0001",
                "CacheClusterId": "my-redis-cluster-0001-002"
            }
        ]
    },
    "ConfigurationEndpoint": {
        "Port": 6379,
        "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "my-redis-cluster",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "07:30-08:30",
    "MemberClusters": [
        "my-redis-cluster-0001-001",
        "my-redis-cluster-0001-002"
    ],
    "CacheNodeType": "cache.t2.micro",
    "DataTiering": "disabled"
    "PendingModifiedValues": {}
}
}

```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou `o--apply-immediately`, verifique o status do cluster usando o `AWS CLI describe-cache-clusters` comando com o parâmetro a seguir. Quando o status mudar para `available`, você poderá começar a usar o novo nó de cluster menor.

Reduzindo a escala de clusters OSS Valkey ou Redis (API) ElastiCache

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando a ElastiCache API. Durante esse processo, o cluster do Valkey ou Redis OSS continuará a atender solicitações com tempo de inatividade mínimo.

O tempo necessário para reduzir a escala verticalmente até um tipo de nó menor varia, dependendo do tipo de nó e da quantidade de dados no seu cluster atual.

Redução de escala (ElastiCache API)

1. Determine quais tipos de nós você pode reduzir usando a `ListAllowedNodeTypeModifications` ação da ElastiCache API com o parâmetro a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação. Use esse parâmetro para descrever um determinado grupo de replicação em vez de todos os grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) a Amazon ElastiCache API Reference.

2. Reduza seu grupo de replicação atual para o novo tipo de nó usando a ação da `ModifyReplicationGroup` ElastiCache API e com os parâmetros a seguir.
 - `ReplicationGroupId` - o nome do grupo de replicação.
 - `CacheNodeType`: o novo tipo de nó menor dos clusters nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela ação `ListAllowedNodeTypeModifications` na etapa anterior.
 - `CacheParameterGroupName` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para

o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.

- `ApplyImmediately`: defina como `true` para fazer com que o processo de redução seja aplicado imediatamente. Para adiar o processo de redução até a próxima janela de manutenção, use `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ApplyImmediately=true  
&CacheNodeType=cache.t2.micro  
&CacheParameterGroupName=redis32-m3-2x1  
&ReplicationGroupId=myReplGroup  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&Version=2014-12-01  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [ModifyReplicationGroup](#) a Amazon ElastiCache API Reference.

Conceitos básicos do filtro Bloom

ElastiCache suporta a estrutura de dados do filtro Bloom, que fornece uma estrutura de dados probabilística eficiente em termos de espaço para verificar se um elemento é membro de um conjunto. Ao usar filtros Bloom, falsos positivos são possíveis: um filtro pode indicar incorretamente que um elemento existe, mesmo que esse elemento não tenha sido adicionado ao conjunto. No entanto, o uso de filtros Bloom evitará falsos negativos, indicações incorretas de que um elemento não existe, mesmo que esse elemento tenha sido adicionado ao conjunto.

Você pode definir a porcentagem de possíveis falsos positivos como uma taxa preferencial para sua workload, ajustando a taxa de fp. Você também pode configurar a capacidade (o número de itens que um filtro Bloom pode conter), propriedades escaláveis e não escaláveis e muito mais.

Depois que você criar um cluster com uma versão de mecanismo compatível, o tipo de dados do Bloom e os comandos associados estarão disponíveis automaticamente. O tipo de dados bloom é compatível com a API com a sintaxe do comando de filtro Bloom das bibliotecas de cliente oficiais da Valkey, incluindo `valkey-py`, `valkey-java` e `valkey-go`. Você pode migrar facilmente os aplicativos Valkey e Redis OSS existentes baseados em Bloom para o ElastiCache. Para obter uma lista completa de comandos, consulte [Comandos de filtro Bloom](#).

As métricas relacionadas ao `BloomBloomFilterBasedCmds`, `BloomFilterBasedCmdsLatency`, e `BloomFilterBasedCmdsECPUs` são incorporadas CloudWatch para monitorar o uso desse tipo de dados. Para obter mais informações, consulte [Métricas para o Valkey e Redis OSS](#).

 Note

- Para usar os filtros Bloom, você deve estar executando o ElastiCache Valkey 8.1 e versões posteriores.
- O tipo de dados bloom não é compatível com RDB com outras ofertas de bloom não baseadas em Valkey.

Visão geral do tipo de dados do filtro Bloom

Os filtros Bloom são uma estrutura de dados probabilística eficiente em termos de espaço que permite adicionar elementos e verificar se existem elementos. Falsos positivos são possíveis quando um filtro indica incorretamente que um elemento existe, mesmo que ele não tenha sido adicionado. No entanto, os filtros Bloom garantem que falsos negativos (indicando incorretamente que um elemento não existe, mesmo que tenha sido adicionado) não ocorram.

A principal fonte de documentação dos filtros bloom pode ser encontrada na página de documentação do [valkey.io](#). Ela contém as seguintes informações:

- [Casos de uso comuns para filtros bloom](#)
 - Desduplicação de anúncios e eventos
 - Detecção de fraudes
 - Filtro de conteúdo nocivo/spam
 - Detecção exclusiva de usuários
- [Diferenças entre filtros bloom escaláveis e não escaláveis](#)
 - Como decidir entre filtros bloom escaláveis e não escaláveis

- [Propriedades do Bloom](#)

- Saiba mais sobre as propriedades ajustáveis dos filtros Bloom. Isso inclui a taxa de falsos positivos, a capacidade, as propriedades de escalabilidade e não escalabilidade e muito mais.

- [Desempenho dos comandos bloom](#)

- [Monitoração das estatísticas gerais do filtro bloom](#)

- [Manipulação de grandes filtros bloom](#)

- Recomendações e detalhes sobre como verificar se um filtro bloom está atingindo seu limite de uso de memória e se ele pode ser dimensionado para atingir a capacidade desejada.
- É possível verificar a quantidade de memória usada por um documento do filtro bloom usando o comando [BF.INFO](#).

Limite de tamanho do bloom

O consumo de memória por um único objeto de filtro Bloom é limitado a 128 MB. É possível verificar a quantidade de memória consumida por um filtro JSON usando o comando `BF.INFO <key> SIZE`.

Bloom ACLs

Semelhante às categorias existentes por tipo de dados (`@string`, `@hash` etc.), uma nova categoria `@bloom` foi adicionada para simplificar o gerenciamento do acesso a comandos e dados do JSON. Nenhum outro comando do Valkey ou Redis OSS existente é membro da categoria `@bloom`.

Existem 3 categorias existentes de ACL que são atualizadas para incluir os novos comandos Bloom: `@read`, `@write` e `@fast`. A tabela a seguir indica o mapeamento de comandos Bloom para as categorias apropriadas.

Comando Bloom	@read	@write	@fast	@bloom
BF.ADD		y	y	y
BF.CARD	y		y	y
BF.EXISTS	y		y	y
BF.INFO	y		y	y

Comando Bloom	@read	@write	@fast	@bloom
BF.INSERT		y	y	y
BF.MADD		y	y	y
BF.MEXISTS	y		y	y
BF.RESERVE	y		y	y

Métricas relacionadas ao filtro Bloom

As seguintes CloudWatch métricas relacionadas às estruturas de dados do Bloom são fornecidas:

Métricas do CW	Unidade	Tecnologia sem servidor/baseado em nós	Description
BloomFilterBasedCommands	Contagem	Ambos	O número total de comandos Bloom, incluindo comandos de leitura e gravação.
BloomFilterBasedCommandsLatency	Microssegundos	Autogerenciado	Latência de todos os comandos de filtro Bloom, incluindo comandos de leitura e gravação.
BloomFilterBasedCommandsECPUs	Contagem	Sem servidor	ECPUs consumido por todos os comandos do filtro Bloom, incluindo comandos de leitura e gravação.

Comandos de filtro Bloom

Os [comandos de filtro Bloom](#) estão documentados no site Valkey.io. Cada página de comando fornece uma visão geral abrangente dos comandos bloom, incluindo sua sintaxe, comportamento, valores de retorno e possíveis condições de erro.

Name (Nome)	Description
BF.ADD	Adiciona um único item a um filtro bloom. Se o filtro ainda não existir, ele será criado.
BF.CARD	Retorna a cardinalidade de um filtro bloom.
BF.EXISTS	Determina se o filtro bloom contém o item especificado.
BF.INFO	Retorna informações de uso e propriedades de um filtro bloom específico.
BF.INSERT	Cria um filtro bloom com 0 ou mais itens ou adiciona itens a um filtro bloom existente.
BF.MADD	Adiciona um ou mais itens a um filtro bloom.
BF.MEXISTS	Determina se o filtro bloom contém 1 ou mais itens.
BF.RESERVE	Cria um filtro bloom vazio com as propriedades especificadas.

Note

O BF.LOAD não é suportado pelo ElastiCache. É relevante apenas para o uso do AOF, que ElastiCache não oferece suporte.

Introdução ao Watch em Serverless

ElastiCache suporta o WATCH comando, que permite monitorar as chaves em busca de alterações e executar [transações](#) condicionais. O WATCH comando é particularmente útil para aplicativos que exigem controle otimista de simultaneidade, garantindo que as transações sejam executadas somente se as chaves monitoradas não tiverem sido modificadas. Isso inclui modificações feitas por um cliente, como comandos de gravação, e pelo próprio Valkey, como expiração ou despejo. Se as chaves forem modificadas a partir do momento em que foram configuradas WATCH e no momento em EXEC que foram recebidas, toda a transação será abortada.

Para ElastiCache Serverless, as seguintes restrições são introduzidas -

ElastiCache O Serverless WATCH tem como escopo um único slot de hash. Isso significa que somente as chaves mapeadas para o mesmo slot de hash podem ser monitoradas ao mesmo tempo pela mesma conexão, e a transação que segue os comandos de observação só pode operar no mesmo slot de hash. Quando um aplicativo tenta monitorar chaves de diferentes slots de hash ou executar comandos de transação que operam em chaves mapeadas para um slot de hash diferente do das chaves monitoradas, um CROSSSLOT erro será retornado. [As tags de hash](#) podem ser usadas para garantir que várias chaves sejam mapeadas no mesmo slot de hash.

Além disso, SCAN o comando não pode ser executado dentro de uma conexão com teclas monitoradas e retornará command not supported during watch state um erro.

A transação será abortada (como se as teclas monitoradas tivessem sido tocadas) quando a ElastiCache Serverless não tiver certeza se uma chave foi modificada. Por exemplo, quando um slot foi migrado e as chaves monitoradas não podem ser encontradas no mesmo nó.

Exemplos de código

Observe e opere teclas de diferentes slots

No exemplo a seguir, a chave monitorada e a chave especificada no SET comando mapeiam para diferentes slots de hash. A execução retorna umCROSSSLOT ERROR.

```
> WATCH foo:{005119}  
OK  
> MULTI  
OK  
> SET bar:{011794} 1234  
QUEUED
```

```
> EXEC
CROSSSLOT Keys in request don't hash to the same slot
```

Observe e opere as teclas do mesmo slot

O exemplo a seguir mostra uma transação bem-sucedida, pois a chave definida WATCH não foi alterada.

```
> WATCH foo:{005119}
OK
> MULTI
OK
> SET bar:{005119} 1234
QUEUED
> EXEC
1) OK
```

Observe as teclas de diferentes slots

No exemplo a seguir, uma tentativa de WATCH digitar chaves de slots diferentes simultaneamente na mesma conexão de cliente retorna CROSSSLOT ERROR a.

```
> WATCH foo:{005119}
OK
> WATCH bar:{123455}
CROSSSLOT Keys in request don't hash to the same slot
```

Limite do relógio

Cada conexão de cliente pode assistir até 1000 teclas ao mesmo tempo.

Comandos suportados relacionados ao Watch

Os comandos [WATCH](#) e [UNWATCH](#) estão documentados no site [Valkey.io](#). Ele fornece uma visão geral abrangente dos comandos, incluindo sua sintaxe, comportamento, valores de retorno e possíveis condições de erro.

Conceitos básicos de pesquisa vetorial

O Amazon ElastiCache for Valkey oferece suporte à pesquisa vetorial, permitindo que você armazene, pesquise e atualize bilhões de incorporações vetoriais de alta dimensão na memória com latências de até microssegundos e recuperação superior a 99%. ElastiCache for Valkey fornece recursos para indexar, pesquisar e atualizar bilhões de incorporações vetoriais de alta dimensão de fornecedores populares como Amazon Bedrock, SageMaker Amazon, Anthropic ou OpenAI para busca e recuperação rápidas. A pesquisa vetorial da Amazon ElastiCache é ideal para casos de uso em que desempenho e escalabilidade máximos são os critérios de seleção mais importantes. Isso inclui geração cache semântico, aumentada de recuperação, recomendações em tempo real, personalização e detecção de anomalias.

A pesquisa vetorial pode ser usada em conjunto com outros ElastiCache recursos para aprimorar seus aplicativos. A pesquisa vetorial por ElastiCache está disponível na versão 8.2 do Valkey em clusters baseados em nós em todas as [AWS regiões](#), sem custo adicional. Para começar, crie um novo cluster Valkey 8.2 usando o [Console de gerenciamento da AWS](#), AWS SDK ou AWS CLI. Também é possível usar a pesquisa vetorial no cluster existente atualizando de qualquer versão anterior do Valkey ou Redis OSS para o Valkey 8.2 em [alguns cliques, sem tempo de inatividade](#).

Visão geral sobre a pesquisa vetorial

ElastiCache for Valkey fornece recursos para indexar, pesquisar e atualizar bilhões de incorporações vetoriais de alta dimensão. A pesquisa vetorial permite criar, manter e usar índices secundários para uma pesquisa eficiente e escalável. Cada operação de pesquisa vetorial se aplica a um único índice. As operações de índice se aplicam somente ao índice especificado. Qualquer número de operações pode ser emitido contra qualquer índice a qualquer momento, com exceção das operações de criação e exclusão de índices. No nível do cluster, várias operações em vários índices podem estar em andamento simultaneamente.

Neste documento, os termos chave, linha e registro são idênticos e usados de maneira intercambiável. Da mesma forma, os termos coluna, campo, caminho e membro também são usados de maneira intercambiável.

O comando `FT.CREATE` pode ser usado para criar um índice para um subconjunto de chaves com os tipos de índice especificados. O comando `FT.SEARCH` realiza consultas nos índices criados e `FT.DROPINDEX` remove um índice existente e todos os dados associados. Não há comandos especiais para adicionar, excluir ou modificar dados indexados. Os comandos `HASH` ou `JSON` existentes que modificam uma chave em um índice atualizam automaticamente esse índice.

Tópicos

- [Índices e o espaço de chaves do Valkey OSS](#)
- [Tipos de campos de índice](#)
- [Algoritmos de índice vetorial](#)
- [Segurança da pesquisa vetorial](#)

Índices e o espaço de chaves do Valkey OSS

Índices são construídos e mantidos em um subconjunto do espaço de chaves do Valkey OSS. O espaço de chaves para cada índice é definido por uma lista de prefixos de chave que são fornecidos quando esse índice é criado. A lista de prefixos é opcional e, se omitida, todo o espaço de chaves fará parte desse índice. Vários índices podem escolher subconjuntos separados ou sobrepostos do espaço de chaves, sem limitação.

Os índices também são digitados, pois cobrem apenas as chaves que têm um tipo correspondente. Atualmente, apenas há suporte para índices dos tipos JSON e HASH. Um índice HASH indexa somente as chaves HASH cobertas por sua lista de prefixos e, da mesma maneira, um índice JSON indexa somente as chaves JSON cobertas por sua lista de prefixos. As chaves na lista de prefixos do espaço de chaves de um índice que não têm o tipo designado são ignoradas e não afetam as operações de pesquisa.

Um índice é atualizado quando um comando modifica qualquer chave que está dentro de um espaço de chaves de um índice. O Valkey extrai automaticamente os campos declarados para cada índice e atualiza o índice com o novo valor. O processo de atualização tem três etapas. Na primeira etapa, a chave HASH ou JSON é modificada e o cliente solicitante é bloqueado. A segunda etapa é executada em segundo plano e atualiza cada um dos índices que contêm a chave modificada. Na terceira etapa, o cliente é desbloqueado. Assim, para operações de consulta realizadas na mesma conexão de uma mutação, essa alteração é imediatamente visível nos resultados da pesquisa.

A criação de um índice é um processo em várias etapas. A primeira delas é executar o comando FT.CREATE, que define o índice. A execução bem-sucedida de uma criação inicia automaticamente a segunda etapa: o preenchimento. O processo de preenchimento é executado em um thread em segundo plano e verifica o espaço de chaves em busca de chaves que estejam na lista de prefixos do novo índice. Cada chave encontrada é adicionada ao índice. Por fim, todo o espaço de chaves é verificado, concluindo o processo de criação do índice. Enquanto o processo de preenchimento está em execução, são permitidas mutações das chaves indexadas, não há restrições, e o processo de preenchimento do índice não será concluído até que todas as chaves estejam devidamente

indexadas. Tentativas de operações de consulta feitas enquanto um índice está sendo preenchido não são permitidas e serão encerradas com um erro. O comando FT.INFO retorna o status do processo de preenchimento no campo 'backfill_status'.

Tipos de campos de índice

Cada índice tem um tipo específico que é declarado quando esse índice é criado junto com um local de um campo (coluna) para ser indexado. Para chaves HASH, a localização é o nome do campo dentro do HASH. Para chaves JSON, a localização é uma descrição do caminho do JSON. Quando uma chave é modificada, os dados associados aos campos declarados são extraídos, convertidos no tipo declarado e armazenados no índice. Se os dados estiverem ausentes ou não puderem ser convertidos com êxito no tipo declarado, esse campo será omitido do índice. Há três tipos de campos, conforme explicado a seguir:

- Campos vetoriais contêm um vetor de números, também conhecido como incorporação de vetores. Os campos vetoriais podem ser usados para filtrar vetores com base em métricas de distância especificadas que medem a similaridade. No caso de índices HASH, o campo deve conter todo o vetor codificado em formato binário (little-endian IEEE 754). No caso de chaves JSON, o caminho deve fazer referência a uma matriz do tamanho correto preenchida com números. Quando uma matriz JSON é usada como um campo vetorial, a representação interna dessa matriz na chave JSON é convertida no formato exigido pelo algoritmo selecionado, reduzindo o consumo e a precisão da memória. Operações de leitura subsequentes usando os comandos JSON produzirão o valor de precisão reduzido.
- Campos numéricos contêm um único número. Campos numéricos podem ser usados com o operador de pesquisa de intervalo. Para HASH, espera-se que o campo contenha o texto ASCII de um número escrito no formato padrão para números fixos ou de pontos flutuantes. Para campos JSON, devem ser seguidas as regras numéricas de números JSON. Independentemente da representação na chave, esse campo é convertido em um número de pontos flutuantes de 64 bits para armazenamento no índice. Como os números subjacentes são armazenados em ponto flutuante com suas limitações de precisão, as regras comuns sobre comparações numéricas para números de pontos flutuantes são aplicáveis.
- Campos de etiqueta contêm zero ou mais valores de etiqueta codificados como uma única string UTF-8. Campos de etiquetas podem ser usados para filtrar consultas de equivalência de valores de etiquetas com comparação com ou sem distinção entre maiúsculas e minúsculas. A string é analisada em valores de etiquetas usando um caractere separador (o padrão é uma vírgula, mas pode ser substituído) com espaços em branco à esquerda e à direita removidos. Qualquer número de valores de etiquetas pode estar contido em um único campo de etiqueta.

Algoritmos de índice vetorial

Dois algoritmos de índice vetorial são compatíveis no Valkey:

- Flat: o algoritmo Flat é um processamento linear por força bruta de cada vetor no índice, produzindo respostas exatas dentro dos limites da precisão dos cálculos de distância. Devido ao processamento linear do índice, os tempos de execução desse algoritmo podem ser muito altos para índices grandes. Índices planos são compatíveis com velocidades de ingestão mais altas.
- HNSW (Hierarchical Navigable Small Worlds): o algoritmo HNSW é uma alternativa que fornece uma aproximação da resposta mais próxima em troca de tempos de execução substancialmente menores. O algoritmo é controlado por três parâmetros: M, EF_CONSTRUCTION e EF_RUNTIME. Os dois primeiros parâmetros são especificados no momento da criação do índice e não podem ser alterados. O parâmetro EF_RUNTIME tem um valor padrão que é especificado no momento da criação do índice, mas pode ser substituído mais tarde em qualquer operação de consulta individual. Esses três parâmetros interagem para equilibrar o consumo de memória e CPU durante operações de ingestão e consulta, bem como para controlar a qualidade da aproximação de uma pesquisa KNN exata (conhecida como taxa de recall).

No HNSW, o parâmetro M controla o número máximo de vizinhos aos quais cada nó pode se conectar, moldando a densidade do índice. Um M maior, como 32 e superior, produz um gráfico mais conectado, melhorando a velocidade de recuperação e consulta porque existem mais caminhos para alcançar vizinhos relevantes. No entanto, ele aumenta o tamanho do índice, o uso da memória e retarda a indexação. Um M menor, como 8 ou menos, gera um faster-to-build índice menor com menor uso de memória, mas o recall diminui e as consultas podem demorar mais devido ao menor número de conexões.

O parâmetro EF_construction determina quantas conexões candidatas são avaliadas ao criar o índice. Um EF_construction mais alto, como 400 e superior, significa que o indexador considera mais caminhos antes de selecionar vizinhos, resultando em um gráfico que melhora a eficiência da recuperação e da consulta posteriormente, mas ao custo de uma indexação mais lenta e maior uso de CPU e memória durante a construção. Um EF_construction baixo, como 64-120, acelera a indexação e reduz o uso de recursos, mas o gráfico resultante pode reduzir a recuperação e retardar as consultas, mesmo se o EF_runtime estiver definido como alto.

Por fim, o EF_runtime controla a amplitude da pesquisa durante a consulta, controlando quantos vizinhos candidatos são explorados em tempo de execução. Defini-la como alta aumenta a recuperação e a precisão, mas às custas da latência da consulta e do uso da CPU. Um EF_runtime

baixo torna as consultas mais rápidas e leves, mas com recuperação reduzida. Ao contrário de `M` ou `EF_construction`, esse parâmetro não afeta o tamanho do índice ou o tempo de construção, tornando-o o parâmetro para ajustar as compensações de recuperação versus latência após a criação de um índice.

Ambos os algoritmos de pesquisa vetorial (Flat e HNSW) aceitam um parâmetro opcional `INITIAL_CAP`. Quando especificado, esse parâmetro pré-aloca memória para os índices, resultando na redução da sobrecarga de gerenciamento de memória e no aumento das taxas de ingestão de vetores. Índices planos oferecem melhores velocidades de ingestão do que o HNSW.

Algoritmos de pesquisa vetorial, como o HNSW, podem não lidar de maneira eficiente com a exclusão ou substituição de vetores inseridos anteriormente. O uso dessas operações pode resultar em consumo excessivo de memória de índice, and/or degradando a qualidade de recall. A reindexação é um método para restaurar a recuperação ideal do uso da memória. and/or

Segurança da pesquisa vetorial

Os mecanismos de segurança de [Valkey ACL \(listas de controle de acesso\)](#) para acesso a comandos e dados são estendidos para controlar o recurso de pesquisa. Há suporte total para o controle de ACL de comandos de pesquisa individuais. É fornecida uma nova categoria de ACL, `@search`, e muitas das categorias existentes (`@fast`, `@read`, `@write` etc.) são atualizadas para incluir os novos comandos. Os comandos de pesquisa não modificam os dados de chaves, o que significa que o mecanismo de ACL existente para acesso de gravação é preservado. As regras de acesso para operações HASH e JSON não são modificadas pela presença de um índice. O controle normal de acesso em nível de chave ainda é aplicado a esses comandos.

Comandos de pesquisa com um índice também têm o acesso controlado por meio da ACL. Verificações de acesso são realizadas no nível do índice inteiro, e não no nível por chave. Isso significa que o acesso a um índice será concedido a um usuário somente se este tiver permissão para acessar todas as chaves possíveis na lista de prefixos do espaço de chaves desse índice. Em outras palavras, o conteúdo real de um índice não controla o acesso. Pelo contrário, é o conteúdo teórico de um índice, conforme definido pela lista de prefixos, que é usado para a verificação de segurança. Situações em que um usuário tem acesso de leitura e and/or gravação a uma chave, mas não consegue acessar um índice contendo essa chave, são possíveis. Observe que somente o acesso para leitura ao espaço de chaves é necessário para criar ou usar um índice: a presença ou ausência do acesso para gravação não é levada em consideração.

Atributos e limites da pesquisa vetorial

Disponibilidade da pesquisa vetorial

A pesquisa vetorial da Amazon ElastiCache está disponível com a versão 8.2 do Valkey em clusters baseados em nós em todas as AWS regiões, sem custo adicional. Você também pode usar a pesquisa vetorial nos clusters existentes atualizando de qualquer versão do Valkey ou do Redis OSS para o Valkey 8.2, em [alguns cliques, sem tempo de inatividade](#).

Atualmente, a pesquisa vetorial está disponível em todos os tipos de ElastiCache instância, exceto nós com classificação de dados em camadas. O uso da pesquisa vetorial em instâncias t2, t3 e t4g exige aumentar a reserva de memória para pelo menos 50% para instâncias micro e 30% para instâncias pequenas. Consulte [esta página](#) para saber mais.

Restrições paramétricas

A tabela a seguir mostra os limites de vários itens de pesquisa vetorial:

Limites da pesquisa vetorial

Item	Valor máximo
Número de dimensões em um vetor	32768
Número de índices que podem ser criados	10
Número de campos em um índice	50
Cláusula TIMEOUT de FT.SEARCH (milissegundos)	60000
Número máximo de prefixos permitidos por índice	16
O tamanho máximo de um campo de tag	10000
O tamanho máximo de um campo numérico	256
Parâmetro HNSW M	2000000
Parâmetro HNSW EF_CONSTRUCTION	4096

Item	Valor máximo
Parâmetro HNSW EF_RUNTIME	4096

Restrições operacionais

Persistência e preenchimento de índices

O processo de atualização tem três etapas. Na primeira etapa, a chave HASH ou JSON é modificada e o cliente solicitante é bloqueado. A segunda etapa é executada em segundo plano e atualiza cada um dos índices que contêm a chave modificada. Na terceira etapa, o cliente é desbloqueado. Assim, para operações de consulta realizadas na mesma conexão de uma mutação, essa alteração é imediatamente visível nos resultados da pesquisa. No entanto, a inserção ou atualização de uma chave pode não ficar visível nos resultados da pesquisa para outros clientes por um curto período. Durante períodos de grande carga do sistema (mutação and/or intensa de dados), o atraso na visibilidade pode se tornar maior.

O atributo de pesquisa vetorial mantém a definição de índices e o conteúdo dos índices. Os índices para campos vetoriais são salvos, mas os índices para TAGS e NUMERIC não são salvos, o que significa que eles devem ser reconstruídos quando carregados externamente (sincronização completa ou recarga). Isso significa que, durante qualquer solicitação ou evento operacional que faça com que um nó seja iniciado ou reiniciado, a definição e o conteúdo do índice para vetores são restaurados com base no snapshot mais recente. Nenhuma ação do usuário é necessária para iniciar isso. No entanto, para índices TAGS e NUMERIC, a reconstrução é executada como uma operação de preenchimento assim que os dados são restaurados. Isso é funcionalmente equivalente ao sistema executar automaticamente um comando FT.CREATE para cada índice definido. Observe que o nó fica disponível para operações do aplicativo assim que os dados são restaurados, mas provavelmente antes da conclusão do preenchimento do índice, o que significa que as operações de preenchimentos voltarão a ficar visíveis para as aplicações.

A conclusão do preenchimento do índice não é sincronizada entre um primário e uma réplica. Essa falta de sincronização pode se tornar inesperadamente visível para as aplicações e, portanto, é recomendável que estas verifiquem a conclusão do preenchimento nos primários e em todas as réplicas antes de iniciar as operações de pesquisa.

Limites de escala

Durante eventos de escalabilidade, o índice pode ser preenchido conforme os dados são migrados. Isso resultará em um recall reduzido para consultas de pesquisa.

Instantâneo import/export e migração ao vivo

Os arquivos RDB de um cluster com índices de pesquisa podem ser importados para outro cluster ElastiCache Valkey com a versão 8.2 ou superior. O novo cluster reconstruirá o conteúdo do índice ao carregar o arquivo RDB. No entanto, a presença de índices de pesquisa em um arquivo RDB limita a compatibilidade desses dados com versões anteriores do Valkey. O formato dos índices de pesquisa definidos pela funcionalidade de pesquisa vetorial só é compreendido por outro ElastiCache cluster com Valkey versão 8.2 ou superior. No entanto, arquivos RDB que não contêm índices não são restritos dessa maneira.

Sem memória durante o preenchimento

Semelhante às operações de gravação do Valkey OSS, um preenchimento de índice está sujeito a limitações. out-of-memory Se a memória do mecanismo ficar cheia enquanto um preenchimento estiver em andamento, todos os preenchimentos serão pausados. Se a memória ficar disponível, o processo de preenchimento será retomado. É possível excluir um índice quando o preenchimento é pausado devido à falta de memória.

Transações

Os comandos `FT.CREATE`, `FT.DROPINDEX`, `FT.ALIASADD`, `FT.ALIASDEL` e `FT.ALIASUPDATE` não podem ser executados em um contexto transacional, ou seja, não dentro de um bloco `MULTI/EXEC` ou dentro de um script `LUA` ou `FUNCTION`.

Escolher a configuração apropriada

Na experiência do console, ElastiCache oferece uma maneira fácil de escolher o tipo de instância certo com base nos requisitos de memória e CPU da sua carga de trabalho vetorial.

Consumo de memória

O consumo de memória é baseado no número de vetores, no número de dimensões, no valor `M` e na quantidade de dados não vetoriais, como metadados associados ao vetor ou outros dados armazenados na instância. A memória total necessária é uma combinação do espaço necessário para os dados vetoriais reais e o espaço necessário para os índices de vetores. O espaço necessário para dados vetoriais é calculado medindo a capacidade real necessária para armazenar vetores em

estruturas de dados HASH ou JSON e a sobrecarga até as placas de memória mais próximas, para alocações de memória ideais. Cada um dos índices de vetores usa referências aos dados vetoriais armazenados nessas estruturas de dados, bem como uma cópia adicional do vetor no índice. É aconselhável planejar esse consumo adicional de espaço pelo índice.

O número de vetores depende de como você decide representar os dados como vetores. Por exemplo, você pode optar por representar um único documento em vários blocos, onde cada um representa um vetor. Como alternativa, você pode optar por representar o documento inteiro como um único vetor. O número de dimensões dos vetores depende do modelo de incorporação escolhido. Por exemplo, se você optar por usar o modelo de incorporação AWS Titan, o número de dimensões seria 1536. Observe que você deve testar o tipo de instância para garantir que ela atenda aos seus requisitos.

Escalar a workload

A pesquisa vetorial é compatível com todos os três métodos de dimensionamento: horizontal, vertical e réplicas. Ao escalar a capacidade, a pesquisa vetorial se comporta exatamente como o Valkey normal, ou seja, aumentar a memória de nós individuais (escalabilidade vertical) ou aumentar o número de nós (escalabilidade horizontal) aumentará a capacidade geral. No modo de cluster, o comando `FT.CREATE` pode ser enviado para qualquer nó primário do cluster e o sistema distribuirá automaticamente a nova definição de índice para todos os membros do cluster.

No entanto, do ponto de vista do desempenho, a pesquisa vetorial se comporta de maneira muito diferente do Valkey normal. A implementação multissegmentada da pesquisa vetorial significa um CPUs rendimento adicional de até aumentos lineares na taxa de transferência de consulta e ingestão. A escalabilidade horizontal gera aumentos lineares no throughput de ingestão, mas pode reduzir o throughput de consultas. Se for necessária uma taxa de transferência adicional de consultas, é necessário escalar por meio de réplicas ou mais CPUs .

Comandos de pesquisa vetorial

Veja a seguir uma lista dos comandos compatíveis com a pesquisa vetorial.

Tópicos

- [FT.CREATE](#)
- [FT.SEARCH](#)
- [FT.DROPINDEX](#)
- [FT.INFO](#)

- [FT._LIST](#)

FT.CREATE

O comando `FT.CREATE` cria um índice vazio e inicia o processo de preenchimento. Cada índice consiste em várias definições de campo. Cada definição de campo especifica um nome de campo, um tipo de campo e um caminho dentro de cada chave indexada para localizar um valor do tipo declarado. Algumas definições de tipo de campo têm especificadores de subtipo adicionais.

Para índices em chaves `HASH`, o caminho é igual ao nome do membro do hash. A cláusula `AS` opcional pode ser usada para renomear o campo, se desejado. A renomeação de campos é especialmente útil quando o nome do membro contém caracteres especiais.

Em índices em chaves `JSON`, o caminho é um caminho `JSON` para os dados do tipo declarado. Como o caminho `JSON` sempre contém caracteres especiais, a cláusula `AS` é obrigatória.

Sintaxe

```
FT.CREATE <index-name>
ON HASH | JSON
[PREFIX <count> <prefix1> [<prefix2>...]]
SCHEMA
(<field-identifier> [AS <alias>]
 | VECTOR [HNSW|FLAT] <attr_count> [<attribute_name> <attribute_value>])
 | TAG [SEPARATOR <sep>] [CASESENSITIVE]
 | NUMERIC
)+
```

`<index-name>` (obrigatório): esse é o nome que você dá ao seu índice. Um erro será retornado se já houver um índice com o mesmo nome.

`ON HASH | JSON` (opcional): somente as chaves que correspondem ao tipo especificado são incluídas nesse índice. Se omitido, o `HASH` é assumido.

`PREFIX <prefix-count> <prefix>` (opcional): se essa cláusula for especificada, somente as chaves que começam com os mesmos bytes de um ou mais dos prefixos especificados serão incluídas nesse índice. Se essa cláusula for omitida, todas as chaves do tipo correto serão incluídas. Um prefixo de comprimento zero também corresponderia a todas as chaves do tipo correto.

Tipos de campo:

- TAG: um campo de tag é uma string que contém um ou mais valores de tag.
 - SEPARATOR <sep> (opcional): um dos caracteres , . <> { } [] " ' : ; ! @ # \$ % ^ & * () - + = ~ usados para delimitar tags individuais. Se for omitido, o padrão será , .
 - CASESENSITIVE (opcional): se presentes, as comparações de tags farão distinção entre maiúsculas e minúsculas. O padrão é que as comparações de tags NÃO façam distinção entre maiúsculas e minúsculas.
- NUMERIC: um campo que contém um número.
- VECTOR: um campo que contém um vetor. Atualmente, dois algoritmos de indexação vetorial são compatíveis: HNSW (Hierarchical Navigable Small World) e FLAT (força bruta). Cada algoritmo tem um conjunto de atributos adicionais, alguns obrigatórios e outros opcionais.
 - FLAT: o algoritmo Flat fornece respostas exatas, mas tem runtime proporcional ao número de vetores indexados e, portanto, pode não ser apropriado para grandes conjuntos de dados.
 - DIM <number> (obrigatório): especifica o número de dimensões em um vetor.
 - TIPO FLOAT32 (obrigatório): Tipo de dados, atualmente somente FLOAT32 é suportado.
 - DISTANCE_METRIC [L2 | IP | COSINE] (obrigatório): especifica o algoritmo de distância.
 - INITIAL_CAP <size> (opcional): tamanho inicial do índice.
 - HNSW: o algoritmo HNSW fornece respostas aproximadas, mas opera substancialmente mais rápido que o FLAT.
 - DIM <number> (obrigatório): especifica o número de dimensões em um vetor.
 - TIPO FLOAT32 (obrigatório): Tipo de dados, atualmente somente FLOAT32 é suportado.
 - DISTANCE_METRIC [L2 | IP | COSINE] (obrigatório): especifica o algoritmo de distância.
 - INITIAL_CAP <size> (opcional): tamanho inicial do índice.
 - M <number> (opcional): número máximo de bordas de saída permitidas para cada nó no gráfico em cada camada. Na camada zero, o número máximo de bordas de saída será 2*M. O padrão é 16, o máximo é 512.
 - EF_CONSTRUCTION <number> (opcional): controla o número de vetores examinados durante a construção do índice. Valores mais altos para esse parâmetro melhorarão a taxa de recall às custas de tempos mais longos de criação do índice. O valor padrão é 200. O valor máximo é 4096.
 - EF_RUNTIME <number> (opcional): controla o número de vetores a serem examinados durante uma operação de consulta. O valor padrão é 10 e o máximo é 4096. Você pode definir esse valor de parâmetro para cada consulta executada. Valores mais altos aumentam os tempos de consulta, mas melhoram a recuperação da consulta.

RESPOSTA: OK ou erro.

FT.SEARCH

Executa uma pesquisa no índice especificado. As chaves que correspondem à expressão da consulta são retornadas.

```
FT.SEARCH <index-name> <query>
[NOCONTENT]
[RETURN <token_count> (<field-identifier> [AS <alias>])+]
[TIMEOUT timeout]
[PARAMS <count> <name> <value> [<name> <value>]]
[LIMIT <offset> <count>]
```

- <index> (obrigatório): o nome do índice que você deseja consultar.
- <query> (obrigatório): a string de consulta, veja abaixo para obter detalhes.
- NOCONTENT (opcional): quando presente, somente os nomes de chave resultantes são retornados, nenhum valor de chave é incluído.
- TIMEOUT <timeout> (opcional): permite definir um valor de tempo limite para o comando de pesquisa. Deve ser um número inteiro em milissegundos.
- PARAMS <count> <name1> <value1> <name2> <value2> ... (opcional): count é do número de argumentos, ou seja, duas vezes o número de pares de nomes de valores. Consulte a string de consulta para obter detalhes de uso.
- RETURN <count> <field1> <field2> ... (opcional): contagem é o número de campos a serem retornados. Especifica os campos que você deseja recuperar dos seus documentos, junto com quaisquer aliases para os valores retornados. Por padrão, todos os campos são retornados, a menos que a opção NOCONTENT seja definida. Nesse caso, nenhum campo será retornado. Se a contagem for definida como 0, ela se comportará da mesma forma que NOCONTENT.
- LIMIT <offset><count>: permite escolher uma parte do resultado. As primeiras <offset> chaves são ignoradas e apenas um máximo de <count> chaves é incluído. O padrão é LIMIT 0 10, que retorna no máximo 10 chaves.
- PARAMS: duas vezes o número de pares de valores-chave. Os key/value pares de parâmetros podem ser referenciados de dentro da expressão de consulta. Para obter mais informações, consulte [Expressão de consulta de pesquisa vetorial](#).
- DIALECT: <dialect> (opcional): especifica seu dialeto. O único dialeto compatível é 2.

RESPONSE

O comando retorna uma matriz em caso de sucesso ou um erro.

Em caso de sucesso, a primeira entrada na matriz de resposta representa a contagem de chaves correspondentes, seguida por uma entrada de matriz para cada chave correspondente. Observe que, se a opção LIMIT for especificada, ela controlará apenas o número de chaves retornadas e não afetará o valor da primeira entrada.

Quando NOCONTENT é especificado, cada entrada na resposta contém somente o nome de chave correspondente. Caso contrário, cada entrada inclui o nome de chave correspondente, seguido por uma matriz dos campos retornados. Os campos de resultado de uma chave consistem em um conjunto de name/value pares. O primeiro name/value par é para a distância calculada. O nome desse par é construído a partir do nome do campo vetorial prefixado com “__” e anexado com “_score” e o valor é a distância calculada. Os name/value pares restantes são os membros e valores da chave, conforme controlado pela RETURN cláusula.

A string de consulta está em conformidade com esta sintaxe:

```
<filtering>=>[ KNN <K> @<vector_field_name> $<vector_parameter_name> <query-modifiers> ]
```

Em que:

- <filtering>: é um * ou uma expressão de filtro. Um * indica que não há filtragem e, portanto, todos os vetores dentro do índice são pesquisados. Uma expressão de filtro pode ser fornecida para designar um subconjunto dos vetores a serem pesquisados.
- <vector_field_name>: o nome de um campo vetorial dentro do índice especificado.
- <K>: o número de vetores vizinhos mais próximos a serem retornados.
- <vector_parameter_name>: um nome de PARAM cujo valor correspondente fornece o vetor de consulta para o algoritmo KNN. Observe que esse parâmetro deve ser codificado como um ponto flutuante binário IEEE 754 de 32 bits no formato little-endian.
- <query-modifiers>: (Opcional) Uma lista de keyword/value pares que modificam essa pesquisa KNN específica. Atualmente, há compatibilidade para duas palavras-chave:
 - EF_RUNTIME: essa palavra-chave é acompanhada por um valor inteiro que substitui o valor padrão de EF_RUNTIME especificado quando o índice foi criado.

- **AS**: essa palavra-chave é acompanhada por um valor de string que se torna o nome do campo de pontuação no resultado, substituindo o algoritmo padrão de geração do nome do campo de pontuação.

Expressão de filtro

Uma expressão de filtro é construída como uma combinação lógica dos operadores de pesquisa de tags e numéricos contidos entre parênteses.

Tag

O operador de pesquisa de tags é especificado com uma ou mais cadeias de caracteres separadas pelo caractere `|`. Uma chave satisfará o operador de pesquisa de tags se o campo indicado contiver qualquer uma das cadeias de caracteres especificadas.

```
@<field_name>:{<tag>}  
or  
@<field_name>:{<tag1> | <tag2>}  
or  
@<field_name>:{<tag1> | <tag2> | ...}
```

Por exemplo, a consulta a seguir retornará documentos com a cor azul OU preta OU verde.

```
@color:{blue | black | green}
```

Como outro exemplo, a consulta a seguir retornará documentos contendo “hello world” ou “hello universe”.

```
@description:{hello world | hello universe}
```

Intervalo numérico

O operador de intervalo numérico permite filtrar consultas para retornar somente valores que estejam entre um determinado valor inicial e final. Tanto consultas de intervalo inclusivas quanto exclusivas são suportadas. Para comparações relacionais simples, `+inf`, `-inf` pode ser usado com uma consulta de intervalo. A sintaxe para um operador de pesquisa de intervalo é:

```
@<field_name>:[ [ ( ] <bound> [ ( ] <bound> ]
```

... em que `<bound>` é um número `+inf` ou `-inf`. Os limites sem um pai aberto principal são inclusivos, enquanto os limites com o pai aberto principal são exclusivos.

Use a tabela a seguir como guia para mapear expressões matemáticas para filtrar consultas:

<code>min <= field <= max</code>	<code>@field:[min max]</code>
<code>min < field <= max</code>	<code>@field:[(min max]</code>
<code>min <= field < max</code>	<code>@field:[min (max]</code>
<code>min < field < max</code>	<code>@field:[(min (max]</code>
<code>field >= min</code>	<code>@field:[min +inf]</code>
<code>field > min</code>	<code>@field:[(min +inf]</code>
<code>field <= max</code>	<code>@field:[-inf max]</code>
<code>field < max</code>	<code>@field:[-inf (max]</code>
<code>field == val</code>	<code>@field:[val val]</code>

Operadores lógicos

Várias tags e operadores de pesquisa numérica podem ser usados para criar consultas complexas usando operadores lógicos.

Lógico AND

Para definir um AND lógico, use um espaço entre os predicados. Por exemplo:

```
query1 query2 query3
```

OR lógico

Para definir um OR lógico, use um espaço entre os predicados. Por exemplo:

```
query1 | query2 | query3
```

Negação lógica

Qualquer consulta pode ser negada colocando o caractere - antes de cada consulta. As consultas negativas retornam todas as entradas que não correspondem à consulta. Isso também inclui chaves que não têm o campo.

Por exemplo, uma consulta negativa em `@genre:{comedy}` retornará todos os livros que não são de comédia E todos os livros que não têm um campo de gênero.

A consulta a seguir retornará todos os livros do gênero “comédia” que não foram publicados entre 2015 e 2024 ou que não têm campo de ano: `@genre:[comedy] - @year:[2015 2024]`

Precedência do operador

As regras típicas de precedência do operador se aplicam, ou seja, NEGATE lógico é a prioridade mais alta, seguido pelo AND lógico e depois OR lógico com a prioridade mais baixa. Os parênteses podem ser usados para substituir a precedência padrão.

Exemplos de combinação de operadores lógicos

Os operadores lógicos podem ser combinados para formar expressões de filtro complexas.

A consulta a seguir retornará todos os livros do gênero “comédia” ou “terror” (AND) publicados entre 2015 e 2024: `@genre:[comedy|horror] @year:[2015 2024]`

A consulta a seguir retornará todos os livros do gênero “comédia” ou “terror” (OR) publicados entre 2015 e 2024: `@genre:[comedy|horror] | @year:[2015 2024]`

A consulta a seguir retornará todos os livros que não têm um campo de gênero ou têm um campo de gênero diferente de “comédia”, publicados entre 2015 e 2024: `-@genre:[comedy] @year:[2015 2024]`

FT.DROPINDEX

Sintaxe

```
FT.DROPINDEX <index-name>
```

O índice especificado é excluído. Retorna OK ou um erro se esse índice não existir.

- `<index-name>` (obrigatório): o nome do índice a ser excluído.

FT.INFO

Sintaxe

```
FT.INFO <index-name>
```

A pesquisa vetorial aumenta o comando [FT.INFO](#) com várias seções adicionais de estatísticas e contadores. Uma solicitação para recuperar a seção SEARCH recuperará todas as estatísticas a seguir:

Nome da chave	Tipo de valor	Description
nome_índice	string	Nome do índice
index_options	string	Reservado. Atualmente definido como "0"
index_definition	array	Veja abaixo a definição desses elementos da matriz.
attributes	matriz de informações de atributos	Um elemento nessa matriz para cada atributo definido, veja abaixo a definição das informações do atributo.
num_docs	integer	Número de chaves atualmente contidas no índice
num_terms	integer	Reservado. Definido atualmente como "0".
record_count	integer	A soma do campo "tamanho" para cada atributo.
hash_indexing_failures	integer	Número de vezes que um atributo não pôde ser convertido para o tipo de atributo declarado. Apesar do nome, isso também se aplica às chaves JSON.
backfill_in_progress	integer	Se um preenchimento estiver em andamento, será um '1', caso contrário, será um '0'
backfill_percent_complete	flutuação	Estimativa da conclusão do preenchimento, um número fracionário na faixa [0..1]

Nome da chave	Tipo de valor	Description
mutation_queue_size	integer	Número de chaves esperando para atualizar o índice.
recent_mutations_queue_delay	integer	Estimativa do atraso (em segundos) da atualização do índice. Será 0 se nenhuma atualização estiver em andamento.
estado	string	Estado de preenchimento: "ready" indica que o preenchimento foi concluído com sucesso. O estado "backfill_in_progress" indica que o preenchimento está em andamento; "backfill_paused_by_oom" significa que o preenchimento foi pausado devido a uma condição de pouca memória. Depois que a condição de pouca memória for resolvida, o preenchimento continuará.

A estrutura `index_definition` é uma matriz de key/value pares definidos como:

Nome da chave	Tipo de valor	Description
key_type	string	Ou a string 'JSON' ou a string 'HASH'
prefixos	array	Cada elemento na matriz é um prefixo definido para o índice. Se nenhum prefixo foi especificado quando o índice

Nome da chave	Tipo de valor	Description
		foi criado, essa matriz terá 0 entradas.
default_score	string	Reservado. Atualmente definido como “1”

Informações do atributo: as informações do atributo são específicas do tipo.

Atributos numéricos:

Chave	Tipo de valor	Description
Identifier	string	Localização do atributo em uma chave. Nome do membro de hash ou caminho JSON
alias	string	Nome do atributo usado nas descrições da consulta.
type	string	A string “NUMERIC”
size	integer	O número de chaves com valores numéricos válidos nesse atributo.

Atributos de tags:

Nome da chave	Tipo de valor	Description
Identifier	string	Localização do atributo em uma chave. Nome do membro de hash ou caminho JSON
alias	string	Nome do atributo usado nas descrições da consulta.

Nome da chave	Tipo de valor	Description
type	string	A string “TAG”
SEPARATOR	caractere	O caractere separador definido quando o índice foi criado
CASESENSITIVE	n/a	Essa chave não tem valor associado. Ela estará presente somente se o atributo tiver sido criado com essa opção.
size	integer	O número de chaves com valores de tag válidos nesse atributo

Atributos de vetores:

Nome da chave	Tipo de valor	Description
Identifier	string	Localização do atributo em uma chave. Nome do membro de hash ou caminho JSON
alias	string	Nome do atributo usado nas descrições da consulta.
type	string	A string “VECTOR”
índice	caractere	Para uma descrição mais detalhada do índice vetorial, veja abaixo.

Descrição do índice vetorial:

Nome da chave	Tipo de valor	Description
capacidade	string	Capacidade atual do índice
dimensões	string	Número de elementos em cada vetor
distance_metric	string	Um dos seguintes: "COSINE", "L2" ou "IP"
size	array	Descrição do índice vetorial, veja abaixo.
data_type	string	Tipo de dados declarado . Somente "FLOAT32" é suportado atualmente.
algoritmo	array	Descrição adicional do algoritmo de pesquisa vetorial.

Descrição do algoritmo de pesquisa vetorial FLAT:

Nome da chave	Tipo de valor	Description
name	string	Nome do algoritmo: FLAT
block_size	número	Tamanho de um bloco do índice FLAT.

Descrição do índice vetorial HNSW:

Nome da chave	Tipo de valor	Description
name	string	Nome do algoritmo: HNSW
m	número	O parâmetro “M” para HNSW

Nome da chave	Tipo de valor	Description
ef_construction	número	O parâmetro "ef_construction" para HNSW
ef_runtime	número	O parâmetro "ef_runtime" para HNSW.

FT._LIST

Liste todos os índices.

Sintaxe

```
FT._LIST
```

Retorna uma matriz de strings que são os nomes do índice atualmente definido.

Conceitos básicos do JSON para Valkey e Redis OSS

O ElastiCache é compatível com o formato JavaScript Object Notation (JSON), que é uma maneira simples e sem esquema de codificar conjuntos de dados complexos em clusters do Valkey e Redis OSS. É possível armazenar e acessar dados nativamente usando o formato JavaScript Object Notation (JSON) dentro dos clusters e atualizar dados do JSON armazenados nesses clusters, sem precisar gerenciar código personalizado para serializá-los e desserializá-los.

Além de usar as operações de API do Valkey e do Redis OSS para aplicações que operam em JSON, agora é possível recuperar e atualizar com eficiência partes específicas de um documento JSON sem precisar manipular todo o objeto. Isso pode melhorar a performance e reduzir os custos. Também é possível pesquisar o conteúdo do seu documento JSON usando a consulta [JSONPath Goessner-style](#) (estilo Goessner).

Depois que você criar um cluster com uma versão de mecanismo compatível, o tipo de dados do JSON e os comandos associados estarão disponíveis automaticamente. Compatível com a API e o RDB com a versão 2 do módulo JSON, para que você possa migrar facilmente as aplicações baseadas em JSON do Valkey e Redis OSS para o ElastiCache. Para obter mais informações sobre os comandos compatíveis, consulte [Comandos compatíveis do Valkey e do Redis OSS](#).

As métricas relacionadas ao JSON `JsonBasedCmds` e `JsonBasedCmdsLatency` são incorporadas ao CloudWatch para monitorar o uso desse tipo de dados. Para obter mais informações, consulte [Métricas do Valkey e Redis OSS](#).

 Note

Para usar o JSON, é necessário estar executando o Valkey 7.2 e posteriores ou Redis OSS 6.2.6 ou posteriores.

Tópicos

- [Visão geral do tipo de dados do JSON](#)
- [Comandos compatíveis do Valkey e do Redis OSS](#)

Visão geral do tipo de dados do JSON

O ElastiCache é compatível com vários comandos do Valkey e do Redis OSS para trabalhar com o tipo de dados do JSON. Veja a seguir uma visão geral do tipo de dados do JSON e uma lista detalhada dos comandos que são compatíveis.

Terminologia

Prazo	Descrição
Documento JSON	Refere-se ao valor de uma chave do JSON.
Valor JSON	Refere-se a um subconjunto de um documento JSON, incluindo a raiz que representa o documento inteiro. Um valor poderia ser um contêiner ou uma entrada dentro de um contêiner.
Elemento JSON	Equivalente ao valor JSON.

Padrão compatível com JSON

O formato JSON é compatível com os padrão de intercâmbio de dados do JSON [RFC 7159](#) e [ECMA-404](#). O padrão UTF-8 [Unicode](#) é compatível com texto do JSON.

Elemento raiz

O elemento raiz pode ser de qualquer tipo de dados do JSON. Observe que na RFC 4627 anterior, somente objetos ou matrizes eram permitidos como valores raiz. Desde a atualização para o RFC 7159, a raiz de um documento JSON pode ser de qualquer tipo de dados do JSON.

Limite de tamanho de documentos

Os documentos JSON são armazenados internamente em um formato que é otimizado para rápido acesso e modificação. Esse formato normalmente resulta no consumo um pouco maior de memória do que a representação serializada equivalente do mesmo documento.

O consumo de memória por um único documento JSON é limitado a 64 MB, que é o tamanho da estrutura de dados na memória, não a string JSON. É possível verificar a quantidade de memória consumida por um documento JSON usando o comando `JSON.DEBUG MEMORY`.

ACLs JSON

- Semelhante às categorias existentes por tipo de dados (`@string`, `@hash` etc.), uma nova categoria `@json` foi adicionada para simplificar o gerenciamento do acesso a comandos e dados do JSON. Nenhum outro comando do Valkey ou Redis OSS existente é membro da categoria `@json`. Todos os comandos JSON impõem restrições e permissões de keypace ou de comando.
- Existem cinco categorias existentes de ACL do Valkey e do Redis OSS que foram atualizadas para incluir os novos comandos JSON: `@read`, `@write`, `@fast`, `@slow` e `@admin`. A tabela a seguir indica o mapeamento de comandos JSON para as categorias apropriadas.

ACL

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		

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

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

Limite de profundidade de aninhamento

Quando um objeto ou matriz JSON tem um elemento que é outro objeto ou matriz JSON, diz-se que esse objeto interno ou matriz se “aninha” dentro do objeto ou matriz externa. O limite máximo de profundidade de aninhamento é 128. Qualquer tentativa de criar um documento que contenha uma profundidade de aninhamento maior que 128 será rejeitada com um erro.

Sintaxe de comando

A maioria dos comandos exige um nome de chave como primeiro argumento. Alguns comandos também têm um argumento path (caminho). O argumento path (caminho) será padronizado para a raiz se for opcional e não fornecido.

Notação:

- Os argumentos obrigatórios são colocados entre colchetes angulares. Por exemplo: <key>
- Os argumentos opcionais são colocados dentro de colchetes. Por exemplo: [path]
- Os argumentos opcionais adicionais são indicados por reticências ("..."). Por exemplo: [json...]

Sintaxe de caminho

O JSON do Redis oferece suporte a dois tipos de sintaxes de path:

- Sintaxe aprimorada: veja abaixo a sintaxe JSONPath descrita por [Goessner](#), conforme mostrado na tabela a seguir. Reordenamos e modificamos as descrições na tabela para maior clareza.
- Sintaxe restrita - Tem recursos de consulta limitados.

Note

Os resultados de alguns comandos são sensíveis ao tipo de sintaxe de caminho usado.

Se um caminho de consulta começar com '\$', ele usará a sintaxe aprimorada. Caso contrário, a sintaxe restrita será usada.

Sintaxe aprimorada

Símbolo/Expressão	Descrição
\$	O elemento raiz.
. ou []	Operador filho.
..	Descida recursiva.

Símbolo/Expressão	Descrição
*	Curinga. Todos os elementos em um objeto ou matriz.
[]	Operador subscrito de matriz. O índice é baseado em 0.
[.]	Operador da união.
[start:end:step]	Operador de matriz slice.
?()	Aplica uma expressão de filtro (script) à matriz ou objeto atual.
()	Expressão de filtro.
@	Usado em expressões de filtro que consultam o nó atual que está sendo processado.
==	Igual a, usado em expressões de filtro.
!=	Não é igual a, usado em expressões de filtro.
>	Maior que, usado em expressões de filtro.
>=	Maior que ou igual a, usado em expressões de filtro.
<	Menor que, usado em expressões de filtro.
<=	Menor que ou igual a, usado em expressões de filtro.
&&	AND lógico, usado para combinar várias expressões de filtro.
	OR lógico, usado para combinar várias expressões de filtro.

Exemplos

Os exemplos a seguir têm como base o exemplo de dados XML de [Goessner](#), que modificamos acrescentando campos adicionais.

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95,  
      "in-stock": true,  
      "sold": true  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99,  
      "in-stock": false,  
      "sold": true  
    },  
    { "category": "fiction",  
      "author": "Herman Melville",  
      "title": "Moby Dick",  
      "isbn": "0-553-21311-3",  
      "price": 8.99,  
      "in-stock": true,  
      "sold": false  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99,  
      "in-stock": false,  
      "sold": false  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95,  
    "in-stock": true,  
    "sold": false  
  }  
}
```



```
}
}
```

Caminho	Descrição
<code>\$.store.book[*].author</code>	Os autores de todos os livros da loja.
<code>\$..author</code>	Todos os autores.
<code>\$.store.*</code>	Todos os membros da loja.
<code>\$["store"].*</code>	Todos os membros da loja.
<code>\$.store..price</code>	O preço de tudo na loja.
<code>\$..*</code>	Todos os membros recursivos da estrutura JSON.
<code>\$..book[*]</code>	Todos os livros.
<code>\$..book[0]</code>	O primeiro livro.
<code>\$..book[-1]</code>	O último livro.
<code>\$..book[0:2]</code>	Os dois primeiros livros.
<code>\$..book[0,1]</code>	Os dois primeiros livros.
<code>\$..book[0:4]</code>	Livros do índice 0 a 3 (o índice final não é inclusivo).
<code>\$..book[0:4:2]</code>	Livros no índice 0, 2.
<code>\$..book[?(@.isbn)]</code>	Todos os livros com um número ISBN.
<code>\$..book[?(@.price<10)]</code>	Todos os livros mais baratos que US\$ 10.
<code>'\$..book[?(@.price < 10)]'</code>	Todos os livros são mais baratos do que US \$ 10. (O caminho deverá estar entre aspas se contiver espaços em branco.)

Caminho	Descrição
'\$..book[?(@["price"] < 10)]'	Todos os livros mais baratos que US\$ 10.
'\$..book[?(@.["price"] < 10)]'	Todos os livros mais baratos que US\$ 10.
'\$..book[?(@.price>=10&&@.price<=100)]'	Todos os livros na faixa de preço entre US\$ 10 a US\$ 100, inclusive.
'\$..book[?(@.price>=10 && @.price<=100)]'	Todos os livros na faixa de preço entre US\$ 10 a US\$ 100, inclusive. (O caminho deverá estar entre aspas se contiver espaços em branco.)
'\$..book[?(@.sold==true @.in-stock==false)]'	Todos os livros vendidos ou esgotados.
'\$..book[?(@.sold == true @.in-stock == false)]'	Todos os livros vendidos ou esgotados. (O caminho deverá estar entre aspas se contiver espaços em branco.)
'\$.store.book[?(@.["category"] == "fiction")]'	Todos os livros na categoria ficção.
'\$.store.book[?(@.["category"] != "fiction")]'	Todos os livros nas categorias não ficção.

Exemplos adicionais de expressões de filtro:

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

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

```
"[3,4,5]"
```

```
127.0.0.1:6379> JSON.SET k3 . '[true,false,true,false,null,1,2,3,4]'
```

```
OK
```

```
127.0.0.1:6379> JSON.GET k3 $.*.[?(@==true)]
```

```
"[true,true]"
```

```
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ == true)]'
```

```
"[true,true]"
```

```
127.0.0.1:6379> JSON.GET k3 $.*.[?(@>1)]
```

```
"[2,3,4]"
```

```
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ > 1)]'
```

```
"[2,3,4]"
```

Sintaxe restrita

Símbolo/Expressão	Descrição
. ou []	Operador filho.
[]	Operador subscrito de matriz. O índice é baseado em 0.

Exemplos

Caminho	Descrição
.store.book[0].author	O autor do primeiro livro.
.store.book[-1].author	O autor do último livro.
.address.city	Nome da cidade.
["store"]["book"][0]["title"]	O título do primeiro livro.
["store"]["book"][-1]["title"]	O título do último livro.

 Note

Todo conteúdo de [Goessner](#) citado nesta documentação está sujeito à [Licença da Creative Commons](#).

Prefixos de erro comuns

Cada mensagem de erro tem um prefixo. Veja a seguir uma lista de prefixos de erro comuns.

Prefixo	Descrição
ERR	Um erro geral.
LIMIT	Um erro que ocorre quando o limite de tamanho é excedido. Por exemplo, o limite de tamanho do documento ou limite de profundidade de aninhamento estava excedido.
NONEXISTENT	Uma chave ou caminho não existe.
OUTOFBOUNDARIES	Índice de matriz fora dos limites.
SYNTAXERR	Erro de sintaxe.
WRONGTYPE	Tipo de valor errado.

Métricas relacionadas ao JSON

As seguintes métricas de informações JSON são fornecidas:

Informações	Descrição
json_total_memory_bytes	Memória total alocada para objetos JSON.
json_num_documents	Número total de documentos no Valkey ou Redis OSS.

Para consultar as métricas principais, execute o seguinte comando:

```
info json_core_metrics
```

Como o ElastiCache para Valkey e Redis OSS interage com JSON

A seção a seguir descreve como o ElastiCache para Valkey e Redis OSS interage com o tipo de dados do JSON.

Precedência do operador

Ao avaliar expressões condicionais para filtragem, `&&`s têm precedência primeiro e, em seguida, `||`s são avaliadas, como é comum na maioria das linguagens. As operações dentro de parênteses são executadas primeiro.

Comportamento do limite máximo de aninhamento de caminho

O limite máximo de aninhamento de caminho no ElastiCache para Redis OSS é 128. Por isso, um valor como `$ . a . b . c . d . . .` só pode atingir 128 níveis.

Processamento de valores numéricos

O JSON não tem tipos de dados separados para números inteiros e de ponto flutuante. Todos eles são chamados de números.

Representações numéricas:

Quando um número JSON é recebido na entrada, ele é convertido em uma das duas representações binárias internas: um valor inteiro com sinal de 64 bits ou um ponto flutuante de fluxo de precisão dupla de IEEE de 64 bits. A string original e toda a sua formatação não serão retidas. Dessa forma, quando um número é gerado como parte de uma resposta JSON, ele é convertido da representação binária interna para uma string imprimível que usa regras genéricas de formatação. Essas regras podem resultar em uma string diferente da que foi recebida.

Comandos aritméticos NUMINCRBY e NUMMULTBY:

- Se ambos os números forem inteiros e o resultado estiver fora da faixa de `int64`, ele se tornará automaticamente um número IEEE de ponto flutuante de precisão dupla de 64 bits.
- Se pelo menos um dos números for um ponto flutuante, o resultado será um número IEEE ponto flutuante de precisão dupla de 64 bits.

- Se o resultado exceder a faixa de 64 bits IEEE dupla, o comando OVERFLOW retornará um erro.

Para obter uma lista detalhada dos comandos disponíveis, consulte [Comandos compatíveis do Valkey e do Redis OSS](#).

Filtragem direta de matriz

O ElastiCache para Valkey ou Redis OSS filtra objetos de matriz diretamente.

Para dados como `[0, 1, 2, 3, 4, 5, 6]` e uma consulta de caminho como `$[?(@<4)]`, ou dados como `{"my_key": [0, 1, 2, 3, 4, 5, 6]}` e uma consulta de caminho como `$.my_key[?(@<4)]`, o ElastiCache retornaria `[1,2,3]` em ambas as circunstâncias.

Comportamento de indexação de matriz

O ElastiCache para Valkey ou Redis OSS permite tanto os índices positivo quanto negativo para matrizes. Para uma matriz de comprimento cinco, 0 consultaria o primeiro elemento, 1 o segundo, e assim por diante. Números negativos começam no fim da matriz, então -1 consultaria o quinto elemento, -2 o quarto elemento e assim por diante.

Para garantir um comportamento previsível para os clientes, o ElastiCache não arredonda os índices de matriz para baixo ou para cima, portanto, se você tiver uma matriz com um comprimento de 5, chamar o índice 5 ou superior, ou -6 ou menor, não produzirá um resultado.

Avaliação estrita da sintaxe

MemoryDB não permite caminhos JSON com sintaxe inválida, mesmo que um subconjunto do caminho contenha um caminho válido. Isso acontece para manter o comportamento correto para nossos clientes.

Comandos compatíveis do Valkey e do Redis OSS

O ElastiCache é compatível com os seguintes comandos JSON do Valkey e do Redis OSS:

Tópicos

- [JSON.ARRAPPEND](#)
- [JSON.ARRINDEX](#)
- [JSON.ARRINSERT](#)

- [JSON.ARRLEN](#)
- [JSON.ARRPOP](#)
- [JSON.ARRTRIM](#)
- [JSON.CLEAR](#)
- [JSON.DEBUG](#)
- [JSON.DEL](#)
- [JSON.FORGET](#)
- [JSON.GET](#)
- [JSON.MGET](#)
- [JSON.MSET](#)
- [JSON.NUMINCRBY](#)
- [JSON.NUMMULTBY](#)
- [JSON.OBJLEN](#)
- [JSON.OBJKEYS](#)
- [JSON.RESP](#)
- [JSON.SET](#)
- [JSON.STRAPPEND](#)
- [JSON.STRLEN](#)
- [JSON.TOGGLE](#)
- [JSON.TYPE](#)

JSON.ARRAPPEND

Anexa um ou mais valores aos valores da matriz no caminho.

Sintaxe

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

- **chave (obrigatório):** uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- **path (obrigatório):** um caminho JSON.

- json (obrigatório) - O valor JSON a ser anexado à matriz.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- NONEXISTENT Erro se o caminho não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, o novo comprimento da matriz.
- Se vários valores de matriz forem selecionados, o comando retornará o novo comprimento da primeira matriz atualizada.
- Erro WRONGTYPE se o valor no caminho não for uma matriz.
- Erro SYNTAXERR se um dos argumentos de entradas json não for uma string JSON válida.
- NONEXISTENT erro se o caminho não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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



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

JSON.ARRINDEX

Procura a primeira ocorrência de um valor escalar JSON nas matrizes no caminho.

- Erros fora do intervalo são tratados arredondando o índice para o início e o fim da matriz.
- Se início > fim, retorna -1 (não encontrado).

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- json-scalar (obrigatório) - O valor escalar a ser pesquisado. JSON escalar se refere a valores que não são objetos ou matrizes. Ou seja, sstring, número, bbooleano, e nulo são valores escalares.
- início (opcional) - O índice inicial, inclusive. Assumirá o padrão de 0 se não for fornecido.
- fim (opcional) - O índice final, exclusivo. Assumirá o padrão de 0 se não for fornecido, o que significa que o último elemento está incluído. 0 ou -1 significa que o último elemento está incluído.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de números inteiros. Cada valor é o índice do elemento correspondente na matriz no caminho. O valor é -1, se não encontrado.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.

Se o caminho for uma sintaxe restrita:

- Inteiro, o índice do elemento correspondente ou -1 se não for encontrado.

- Erro WRONGTYPE se o valor no caminho não for uma matriz.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.ARRINSERT

Insere um ou mais valores nos valores da matriz no caminho antes do índice.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- índice (obrigatório) -Um índice de matriz antes do qual os valores são inseridos.
- json (obrigatório) - O valor JSON a ser anexado à matriz.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor for uma matriz vazia, seu valor de retorno correspondente será nulo.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Erro `OUTOFBOUNDARIES` se o argumento índice estiver fora dos limites.

Se o caminho for uma sintaxe restrita:

- Inteiro, o novo comprimento da matriz.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.
- Erro `OUTOFBOUNDARIES` se o argumento índice estiver fora dos limites.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.ARRLEN

Obtém o comprimento dos valores da matriz no caminho.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o comprimento da matriz em cada caminho.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, comprimento da matriz.
- Se vários objetos forem selecionados, o comando retornará o comprimento da primeira matriz.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.
- `NONEXISTENT` JSON erro se o caminho não existir.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

127.0.0.1:6379> JSON.SET k2 . '[[[], "a", ["a", "b"], ["a", "b", "c"], 4]]'
OK
```

```
127.0.0.1:6379> JSON.ARRLEN k2 $[*]  
1) (integer) 0  
2) (nil)  
3) (integer) 2  
4) (integer) 3  
5) (nil)
```

Sintaxe do caminho restrita:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]]'  
OK  
127.0.0.1:6379> JSON.ARRLEN k1 [*]  
(integer) 0  
127.0.0.1:6379> JSON.ARRLEN k1 [1]  
(integer) 1  
127.0.0.1:6379> JSON.ARRLEN k1 [2]  
(integer) 2  
  
127.0.0.1:6379> JSON.SET k2 . '[[[], "a", ["a", "b"], ["a", "b", "c"], 4]'  
OK  
127.0.0.1:6379> JSON.ARRLEN k2 [1]  
(error) WRONGTYPE JSON element is not an array  
127.0.0.1:6379> JSON.ARRLEN k2 [0]  
(integer) 0  
127.0.0.1:6379> JSON.ARRLEN k2 [6]  
(error) OUTOFBOUNDRIES Array index is out of bounds  
127.0.0.1:6379> JSON.ARRLEN k2 a.b  
(error) NONEXISTENT JSON path does not exist
```

JSON.ARRPOP

Remove e retorna elemento no índice da matriz. Exibir uma matriz vazia retorna nulo.

Sintaxe

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

- **chave (obrigatório):** uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- **path (opcional):** um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

- índice (opcional) -A posição na matriz a partir da qual começar a exibir.
 - O padrão é -1 se não é fornecido, o que significa o último elemento.
 - O valor negativo significa posição do último elemento.
 - Os índices fora do limite são arredondados para seus respectivos limites de matriz.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings em massa que representam os valores exibidos em cada caminho.
- Se um valor for uma matriz vazia, seu valor de retorno correspondente será nulo.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.

Se o caminho for uma sintaxe restrita:

- String em massa que representa o valor JSON exibido.
- Nulo se a matriz estiver vazia.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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

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

JSON.ARRTRIM

Reduz uma matriz no caminho para que ela se torne uma subarray [start, end], ambos inclusivos.

- Se a matriz estiver vazia, não faça nada, retorne 0.
- Se início for < 0, trate-a como 0.
- Se fim for >= tamanho (tamanho da matriz), trate-a como tamanho-1.
- Se início for >= tamanho ou início for > fim, esvazie a matriz e retorne 0.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- início (obrigatório) - O índice inicial, inclusive.
- fim (obrigatório) - O índice final, inclusive.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor for uma matriz vazia, seu valor de retorno correspondente será nulo.

- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Erro `OUTOFBOUNDARIES` se um argumento índice estiver fora dos limites.

Se o caminho for uma sintaxe restrita:

- Inteiro, o novo comprimento da matriz.
- Nulo se a matriz estiver vazia.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.
- Erro `OUTOFBOUNDARIES` se um argumento índice estiver fora dos limites.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.CLEAR

Limpa as matrizes ou um objeto no caminho.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

- Inteiro, o número de contêineres limpos.
- Limpar uma matriz ou objeto vazio conta como 1 contêiner limpo.
- Limpar um valor que não seja do contêiner retorna 0.

Examples

```
127.0.0.1:6379> JSON.SET k1 . '[[[], [0], [0,1], [0,1,2], 1, true, null, "d"]]'
OK
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 7
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 4
127.0.0.1:6379> JSON.SET k2 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.CLEAR k2 .children
(integer) 1
127.0.0.1:6379> JSON.GET k2 .children
"[]"
```

JSON.DEBUG

Informações de relatório. Os subcomandos compatíveis são:

- MEMORY <key> [path] - Informa o uso de memória em bytes de um valor JSON. O caminho assumirá o padrão da raiz se não for fornecido.
- FIELDS <key> [path] - Informa o número de campos no caminho do documento especificado. O caminho assumirá o padrão da raiz se não for fornecido. Cada valor JSON não contêiner conta como um campo. Objetos e matrizes contam recursivamente como um campo para cada um

dos valores JSON que contêm. Cada valor de contêiner, exceto o contêiner raiz, conta como um campo adicional.

- HELP - Imprime mensagens de ajuda referentes ao comando.

Sintaxe

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

Depende do subcomando:

MEMORY

- Se o caminho for uma sintaxe aprimorada:
 - Retorna uma matriz de inteiros que representa o tamanho da memória (em bytes) do valor JSON em cada caminho.
 - Retornará uma matriz vazia se a chave do Valkey ou Redis OSS não existir.
- Se o caminho for uma sintaxe restrita:
 - Retorna um número inteiro, tamanho da memória eo valor de JSON em bytes.
 - Retornará nulo se a chave do Valkey ou Redis OSS não existir.

FIELDS

- Se o caminho for uma sintaxe aprimorada:
 - Retorna uma matriz de inteiros que representam o número de campos do valor JSON em cada caminho.
 - Retornará uma matriz vazia se a chave do Valkey ou Redis OSS não existir.
- Se o caminho for uma sintaxe restrita:
 - Retorna um número inteiro, número de campos do valor JSON.
 - Retornará nulo se a chave do Valkey ou Redis OSS não existir.

HELP - Retorna uma série de mensagens de ajuda.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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

```
127.0.0.1:6379> JSON.DEBUG HELP
```

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

JSON.DEL

Exclui valores JSON no caminho em uma chave de documento. Se o caminho é a raiz, é equivalente a excluir a chave do Valkey ou Redis OSS.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

- Número de elementos excluídos.
- 0 se a chave do Valkey ou Redis OSS não existir.
- 0 se o caminho JSON for inválido ou não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

```
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

Sintaxe do caminho restrita:

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

JSON.FORGET

Um alias de [JSON.DEL](#).

JSON.GET

Retorna o JSON serializado em um ou vários caminhos.

Sintaxe

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

- **chave (obrigatório):** uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- **INDENT/NEWLINE/SPACE (opcional)** —Controla o formato da string do JSON retornada, isto é, "formatado para impressão". O valor padrão é uma string vazia. Ele pode ser anulado em qualquer combinação. Eles podem ser especificados em qualquer ordem.

- NOESCAPE - Opcional, permitido estar presente para compatibilidade com legado e não tem outro efeito.
- path (opcional): zero ou mais caminhos JSON, assumirá o padrão de raiz se nenhum for fornecido. Os argumentos do caminho devem ser colocados no final.

Return

Sintaxe do caminho aprimorada:

Se um caminho for fornecido:

- Retornará a string serializada de uma matriz de valores.
- Se nenhum valor for selecionado, o comando retornará uma matriz vazia.

Se vários caminhos forem fornecidos:

- Retornará um objeto JSON em formato de string, no qual cada caminho é uma chave.
- Se houver sintaxe mista de caminho aprimorado e restrito, o resultado estará de acordo com a sintaxe aprimorada.
- Se um caminho não existir, seu valor correspondente será uma matriz vazia.

Exemplos

Sintaxe do caminho aprimorada:

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

```
OK
127.0.0.1:6379> json.get k2 $.*
"[{},{\"a\":1},{\"a\":1,\"b\":2},1,1,2]"
```

Sintaxe do caminho restrita:

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

JSON.MGET

Obtém JSONs serializados no caminho a partir de várias chaves de documento. Ele retorna nulo para uma chave ou caminho JSON não existente .

Sintaxe

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

- chave (obrigatório): uma ou mais chaves do Valkey ou Redis OSS do tipo documento.
- path (obrigatório): um caminho JSON.

Return

- Matriz de strings em massa. O tamanho da matriz é igual ao número de chaves no comando. Cada elemento da matriz será preenchido com (a) o JSON serializado conforme localizado pelo caminho

ou (b)nulo se a chave não existir, o caminho não existir no documento ou o caminho for inválido (erro de sintaxe).

- Se alguma das chaves especificadas existir e não for uma chave JSON, o comando retornará o erro WRONGTYPE.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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


JSON.MSET

Compatível com Valkey versão 8.1 e superior.

Defina valores JSON para várias chaves. A operação é atômica. Todos os valores estão definidos ou nenhum está definido.

Sintaxe

```
JSON.MSET key path json [ key path json ... ]
```

- Se o caminho exigir um membro do objeto:
 - Se o elemento pai não existir, o comando retornará o erro NONEXISTENT.
 - Se o elemento pai existir, mas não for um objeto, o comando retornará ERROR.
 - Se o elemento pai existir e for um objeto:
 - Se o membro não existir, um novo membro será anexado ao objeto pai se e somente se o objeto pai for o último filho no caminho. Caso contrário, o comando retornará o erro NONEXISTENT.
 - Se o membro existir, seu valor será substituído pelo valor JSON.
- Se o caminho exigir um índice de matriz:
 - Se o elemento pai não existir, o comando retornará o erro NONEXISTENT.
 - Se o elemento pai existir, mas não for uma matriz o comando retornará ERROR.
 - Se o elemento pai existir, mas o índice estiver fora dos limites, o comando retornará o erro OUTFOUBOUNDARIES.
 - Se o elemento pai existir e o índice for válido, o elemento será substituído pelo novo valor JSON.
- Se o caminho solicitar um objeto ou matriz, o valor (objeto ou matriz) será substituído pelo novo valor JSON.

Return

- Resposta de sequência de caracteres simples: 'OK' se a operação foi bem-sucedida.
- Resposta de erro simples: se a operação falhar.

Exemplos

Sintaxe do caminho aprimorada:

```
127.0.0.1:6379> JSON.MSET k1 . '[1,2,3,4,5]' k2 . '{"a":{"a":1, "b":2, "c":3}}' k3 .
'{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.GET k1
"[1,2,3,4,5]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.MSET k2 $.a.* '0' k3 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"
127.0.0.1:6379> JSON.GET k3
"{\"a\":[0,0,0,0,0]}"
```

Sintaxe do caminho restrita:

```
127.0.0.1:6379> JSON.MSET k1 . '{"name": "John","address": {"street": "123 Main
St","city": "Springfield"},"phones": ["555-1234","555-5678"]}'
OK
127.0.0.1:6379> JSON.MSET k1 .address.street '"21 2nd Street"' k1 .address.city '"New
York"'
OK
127.0.0.1:6379> JSON.GET k1 .address.street
 "\"21 2nd Street\""
127.0.0.1:6379> JSON.GET k1 .address.city
 "\"New York\""
```

JSON.NUMINCRBY

Incrementa os valores numéricos no caminho por um determinado número.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- número (obrigatório) - Um número.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings em massa que representam os valores resultantes em cada caminho.
- Se um valor não for um número, seu valor de retorno correspondente será nulo.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de duplo IEEE de 64 bits.
- `NONEXISTENT` se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz de strings em massa que representam os valores resultantes em cada caminho.
- Se vários valores forem selecionados, o comando retornará o resultado do último valor atualizado.
- Erro `WRONGTYPE` se o valor no caminho não for um número.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de duplo IEEE de 64 bits.
- `NONEXISTENT` se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

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

```

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

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

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

```

Sintaxe do caminho restrita:

```

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

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'

```

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

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

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

```
"4"
```

JSON.NUMMULTBY

Multiplica os valores numéricos no caminho por um determinado número.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- número (obrigatório) - Um número.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings em massa que representam o valor resultante em cada caminho.
- Se um valor não for um número, seu valor de retorno correspondente será nulo.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de um número IEEE de 64 bits do ponto flutuante de precisão dupla.
- `NONEXISTENT` se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz de strings em massa que representam os valores resultantes em cada caminho.
- Se vários valores forem selecionados, o comando retornará o resultado do último valor atualizado.
- Erro `WRONGTYPE` se o valor no caminho não for um número.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de um número IEEE de 64 bits duplo.
- `NONEXISTENT` se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

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

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

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

```
"[2,null,6]"
```

Sintaxe do caminho restrita:

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

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

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



```

"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"\"a\":2},\"b\":{\"\"a\":2,\"b\":4},\"d\":{\"\"a\":2,\"b\":4,\"c\":6}}}"

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

```

JSON.OBJLEN

Obtém o número de chaves nos valores do objeto no caminho.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o comprimento do objeto em cada caminho.
- Se um valor não for um objeto, seu valor de retorno correspondente será nulo.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, número de chaves no objeto.
- Se vários objetos forem selecionados, o comando retornará o comprimento do primeiro objeto.
- Erro `WRONGTYPE` se o valor no caminho não for um objeto.
- `NONEXISTENT` JSON erro se o caminho não existir.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.OBJKEYS

Obtém nomes de chave nos valores de objeto no caminho.

Sintaxe

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

- **chave (obrigatório):** uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- **path (opcional):** um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de matriz de strings em massa. Cada elemento é uma matriz de chaves em um objeto correspondente.
- Se um valor não for um objeto, seu valor de retorno correspondente será vazio.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz de strings em massa. Cada elemento é um nome de chave no objeto.
- Se vários objetos forem selecionados, o comando retornará as chaves do primeiro objeto.
- Erro `WRONGTYPE` se o valor no caminho não for um objeto.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

```
127.0.0.1:6379> JSON.OBJKEYS k1 .*
1) "a"
127.0.0.1:6379> JSON.OBJKEYS k1 .d
1) "a"
2) "b"
3) "c"
```

JSON.RESP

Retorna o valor JSON em um determinado caminho no protocolo de Serialização do Valkey ou Redis OSS (RESP). Se o valor for contêiner, a resposta será uma matriz RESP ou matriz aninhada.

- JSON nulo é mapeado para o RESP Null Bulk String.
- Valores booleanos JSON são associados às respectivas strings simples RESP.
- Os números inteiros são mapeados para números inteiros RESP.
- Os números de ponto flutuante duplo IEEE de 64 bits são mapeados para RESP Bulk Strings.
- As strings JSON são mapeadas para RESP Bulk Strings.
- As matrizes JSON são representados como matrizes RESP, onde o primeiro elemento é a string simples [, seguida pelos elementos da matriz.
- Os objetos JSON são representados como matrizes RESP, onde o primeiro elemento é a string simples {, seguida por pares de valores-chave, cada um dos quais é uma string RESP em massa.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de matrizes. Cada elemento da matriz representa a forma RESP do valor em um caminho.
- Matriz vazia se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz representa a forma RESP do valor em um caminho.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

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

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

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

- 2) 1) "type"
- 2) "office"
- 3) 1) "number"
- 2) "555 555-4567"

```
127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers[*]
```

- 1) 1) {
 - 2) 1) "type"
 - 2) "home"
- 3) 1) "number"
- 2) "212 555-1234"
- 2) 1) {
 - 2) 1) "type"
 - 2) "office"
- 3) 1) "number"
- 2) "555 555-4567"

Sintaxe do caminho restrita:

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

```
127.0.0.1:6379> JSON.RESP k1 .address
```

- 1) {
- 2) 1) "street"
- 2) "21 2nd Street"
- 3) 1) "city"
- 2) "New York"
- 4) 1) "state"
- 2) "NY"
- 5) 1) "zipcode"
- 2) "10021-3100"

```
127.0.0.1:6379> JSON.RESP k1
```

- 1) {
- 2) 1) "firstName"
- 2) "John"

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

JSON.SET

Define os valores JSON no caminho.

Se o caminho exigir um membro do objeto:

- Se o elemento pai não existir, o comando retornará um erro INEXISTENT.

- Se o elemento pai não existir, o comando retornará ERROR.
- Se o elemento pai existir e for um objeto:
 - Se o membro não existir, um novo membro será anexado ao objeto pai se e somente se o objeto pai for o último filho no caminho. Caso contrário, o comando retornará um erro INEXISTENT.
 - Se o membro existir, seu valor será substituído pelo valor JSON.

Se o caminho exigir um índice de matriz:

- Se o elemento pai não existir, o comando retornará um erro INEXISTENT.
- Se o elemento pai existir, mas não for uma matriz o comando retornará ERROR.
- Se o elemento pai existir, mas o índice estiver fora dos limites, o comando retornará um erro OUTFBOUNDARIES.
- Se o elemento pai existir e o índice for válido, o elemento será substituído pelo novo valor JSON.

Se o caminho solicitar um objeto ou matriz, o valor (objeto ou matriz) será substituído pelo novo valor JSON.

Sintaxe

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

[NX | XX] Onde é possível ter 0 ou 1 de identificadores [NX | XX].

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (obrigatório): um caminho JSON. Para uma nova chave, o caminho JSON deve ser a raiz “.”.
- NX (opcional): se o caminho for a raiz, defina o valor somente se a chave não existir. Isto é, insira um novo documento. Se o caminho não for a raiz, defina o valor somente se a chave Redis não existir. Isto é, insira um valor no documento.
- XX (opcional): se o caminho for a raiz, defina o valor somente se a chave existir. Isto é, substitua o documento existente. Se o caminho não for a raiz, defina o valor somente se a chave Redis existir. Isto é, atualize o valor existente.

Return

- String simples 'OK' em caso de sucesso.

- Nulo se a condição NX ou XX não for atendida.

Exemplos

Sintaxe do caminho aprimorada:

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

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

Sintaxe do caminho restrita:

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

JSON.STRAPPEND

Anexa uma string às strings JSON no caminho.

Sintaxe

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

- **chave (obrigatório):** uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- **path (opcional):** um caminho JSON. Assumirá o padrão da raiz se não for fornecido.
- **json_string (obrigatório)** - A representação JSON de uma string. Observe que uma string JSON deve estar entre aspas. Por exemplo: `"exemplo de string"`.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor não for uma string, seu valor de retorno correspondente será nulo.
- **SYNTAXERREro** se o argumento da entrada json não for uma string JSON válida.
- **NONEXISTENTEro** se o caminho não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, o novo comprimento da string.
- Se vários valores de string forem selecionados, o comando retornará o novo comprimento da última string atualizada.
- Erro **WRONGTYPE** se o valor no caminho não for uma string.
- Erro **WRONGTYPE** se o argumento da entrada json não for uma string JSON válida.
- **NONEXISTENTEro** se o caminho não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

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

Sintaxe do caminho restrita:

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

JSON.STRLLEN

Obtém o comprimento dos valores da string JSON no caminho.

Sintaxe

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

- **chave (obrigatório):** uma chave do Valkey ou Redis OSS do tipo de documento JSON.

- `path` (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o comprimento do valor da string em cada caminho.
- Se um valor não for uma string, seu valor correspondente será nulo.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, o comprimento da string.
- Se vários valores de string forem selecionados, o comando retornará o comprimento da primeira string.
- Erro `WRONGTYPE` se o valor no caminho não for uma string.
- `NONEXISTENT` erro se o caminho não existir.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
```

```
OK
```

```
127.0.0.1:6379> JSON.STRLEN k1 $.a.a
```

```
1) (integer) 1
```

```
127.0.0.1:6379> JSON.STRLEN k1 $.a.*
```

```
1) (integer) 1
```

```
127.0.0.1:6379> JSON.STRLEN k1 $.c.*
```

```
1) (integer) 1
```

```
2) (integer) 2
```

```
127.0.0.1:6379> JSON.STRLEN k1 $.c.b
```

```
1) (integer) 2
```

```
127.0.0.1:6379> JSON.STRLEN k1 $.d.*
```

```
1) (nil)
```

```
2) (integer) 1
```

3) (nil)

Sintaxe do caminho restrita:

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

JSON.TOGGLE

Alterna valores booleanos entre verdadeiro e falso no caminho.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros (0 - falso, 1 - verdadeiro) que representam o valor booleano resultante em cada caminho.
- Se um valor não for um valor booleano, seu valor de retorno correspondente será nulo.
- NONEXISTENT se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- String (“verdadeiro”/“falso”) que representa o valor booleano resultante.
- NONEXISTENT se a chave do documento não existir.
- Erro WRONGTYPE se o valor no caminho não for um valor booleano.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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

```
OK
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"true"
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"false"
```

JSON.TYPE

Informa o tipo de valores em um determinado caminho.

Sintaxe

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

- chave (obrigatório): uma chave do Valkey ou Redis OSS do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings que representam o tipo de valor em cada caminho. O tipo é um destes {"null", "boolean", "string", "number", "integer", "object" e "array"}.
- Se um caminho não existir, seu valor de retorno correspondente será nulo.
- Matriz vazia se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- String, tipo do valor
- Nulo se a chave do documento não existir.
- Nulo se o caminho JSON for inválido ou não existir.

Exemplos

Sintaxe do caminho aprimorada:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
```



```
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
2) number
3) string
4) boolean
5) null
6) object
7) array
```

Sintaxe do caminho restrita:

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

Marcando seus recursos ElastiCache

Para ajudar você a gerenciar seus clusters e outros ElastiCache recursos, você pode atribuir seus próprios metadados a cada recurso na forma de tags. As tags permitem que você categorize seus AWS recursos de maneiras diferentes, por exemplo, por finalidade, proprietário ou ambiente. Isso é

útil quando você tem muitos recursos do mesmo tipo. É possível identificar rapidamente um recurso específico baseado nas tags que você atribuiu a ele. Este tópico descreve tags e mostra a você como criá-las.

 Warning

Como uma prática recomendada, sugerimos que você não inclua dados confidenciais nas suas tags.

Conceitos Básicos de Tags

Uma tag é um rótulo que você atribui a um AWS recurso. Cada tag consiste de uma chave e um valor opcional, que podem ser definidos. As tags permitem que você categorize seus AWS recursos de maneiras diferentes, por exemplo, por finalidade ou proprietário. Por exemplo, você pode definir um conjunto de tags para os ElastiCache clusters da sua conta que ajuda a rastrear o proprietário e o grupo de usuários de cada instância.

Recomendamos que você desenvolva um conjunto de chave de tags que atenda suas necessidades para cada tipo de recurso. Usar um conjunto consistente de chaves de tags facilita para você gerenciar seus recursos. É possível pesquisar e filtrar os recursos de acordo com as tags que adicionar. Para obter mais informações sobre como implementar uma estratégia eficaz de marcação de recursos, consulte o [whitepaper da AWS, Práticas recomendadas de marcação](#).

As tags não têm nenhum significado semântico ElastiCache e são interpretadas estritamente como uma sequência de caracteres. Além disso, as tags não são automaticamente atribuídas aos seus recursos. É possível editar chaves de tags e valores e é possível remover as tags de um recurso a qualquer momento. É possível definir o valor de uma tag como `null`. Ao adicionar uma tag que tenha a mesma chave de uma tag existente nesse recurso, o novo valor substituirá o antigo. Caso exclua um recurso, todas as respectivas tags também serão excluídas. Além disso, se você adicionar ou excluir tags em um grupo de replicação, todos os nós nesse grupo de replicação também terão suas tags adicionadas ou removidas.

Você pode trabalhar com tags usando a Console de gerenciamento da AWS, a AWS CLI, a e a ElastiCache API.

Se você estiver usando o IAM, você pode controlar quais usuários AWS da sua conta têm permissão para criar, editar ou excluir tags. Para obter mais informações, consulte [Permissões em nível de recurso](#).

Recursos que podem ser marcados

Você pode marcar a maioria dos ElastiCache recursos que já existem na sua conta. A tabela a seguir lista os recursos compatíveis com o uso de tags. Se você estiver usando o Console de gerenciamento da AWS, você pode aplicar tags aos recursos usando o [Editor de tags](#). Algumas telas de recursos permitem que você especifique tags para um recurso ao criá-lo; por exemplo, uma tag com uma chave de nome e um valor que você especificar. Na maioria dos casos, o console aplicará as tags imediatamente depois de o recurso ser criado (em vez de durante a criação de recursos). O console pode organizar os recursos de acordo com a tag Name, mas essa tag não tem nenhum significado semântico para o ElastiCache serviço.

Além disso, algumas ações de criação de recursos permitem que você especifique tags para um recurso quando ele é criado. Se as tags não puderem ser aplicadas durante a criação dos recursos, nós reverteremos o processo de criação de recursos. Isso garante que os recursos sejam criados com tags ou, então, não criados, e que nenhum recurso seja deixado sem tags. Ao marcar com tags os recursos no momento da criação, você elimina a necessidade de executar scripts personalizados de uso de tags após a criação do recurso.

Se você estiver usando a ElastiCache API da Amazon, a AWS CLI ou um AWS SDK, poderá usar o Tags parâmetro na ação relevante da ElastiCache API para aplicar tags. Eles são:

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`
- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

A tabela a seguir descreve os ElastiCache recursos que podem ser marcados e os recursos que podem ser marcados na criação usando a ElastiCache API, a AWS CLI ou um AWS SDK.

Suporte de marcação para recursos ElastiCache

Compatível com tags	Oferece suporte à marcação na criação
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim

 Note

Não é possível marcar Datastores globais.

Você pode aplicar permissões em nível de recurso baseadas em tags em suas políticas do IAM às ações da ElastiCache API que oferecem suporte à marcação na criação para implementar um controle granular sobre os usuários e grupos que podem marcar recursos na criação. Seus recursos estão devidamente protegidos a partir da criação. As tags são aplicadas imediatamente aos recursos. Portanto, todas as permissões em nível de recurso baseadas em tags que controlam o uso de recursos entram imediatamente em vigor. Seus recursos podem ser rastreados e relatados com mais precisão. É possível obrigar o uso de marcação com tags nos novos recursos e controlar quais chaves e valores de tag são definidos nos seus recursos.

Para obter mais informações, consulte [Exemplo de marcação de recursos](#).

Para obter mais informações sobre como marcar os seus recursos para o faturamento, consulte [Monitoramento de custos com tags de alocação de custos](#).

Marcação de caches e snapshots

As seguintes regras se aplicam à marcação como parte das operações de solicitação:

- `CreateReplicationGroup`:
 - Se os parâmetros `--primary-cluster-id` e `--tags` forem incluídos na solicitação, as tags de solicitação serão adicionadas ao grupo de replicação e propagadas para todos os clusters no grupo de replicação. Se o cluster primário tiver tags existentes, elas serão substituídas pelas tags de solicitação para ter tags consistentes em todos os nós.

Se não houver tags de solicitação, as tags de cluster primário serão adicionadas ao grupo de replicação e propagadas para todos os clusters.

- Se o `--snapshot-name` ou o `--serverless-cache-snapshot-name` for fornecido:

Se as tags forem incluídas na solicitação, o grupo de replicações será marcado somente com essas tags. Se nenhuma tag for incluída na solicitação, as tags de snapshot serão adicionadas ao grupo de replicações.

- Se o `--global-replication-group-id` for fornecido:

Se as tags forem incluídas na solicitação, as tags de solicitação serão adicionadas ao grupo de replicação e propagadas para todos os clusters.

- **CreateCacheCluster :**

- Se o `--replication-group-id` for fornecido:

Se as tags forem incluídas na solicitação, o cluster será marcado somente com essas tags. Se nenhuma tag for incluída na solicitação, o cluster herdará as tags de grupo de replicação em vez das tags do cluster primário.

- Se o `--snapshot-name` for fornecido:

Se as tags forem incluídas na solicitação, o cluster será marcado somente com essas tags. Se nenhuma tag for incluída na solicitação, as tags de snapshot serão adicionadas ao cluster.

- **CreateServerlessCache :**

- Se as tags forem incluídas na solicitação, somente as tags de solicitação serão adicionadas ao cache sem servidor.

- **CreateSnapshot :**

- Se o `--replication-group-id` for fornecido:

Se as tags forem incluídas na solicitação, somente as tags de solicitação serão adicionadas ao snapshot. Se nenhuma tag for incluída na solicitação, as tags de grupo de replicação serão adicionadas ao cluster de cache.

- Se o `--cache-cluster-id` for fornecido:

Se as tags forem incluídas na solicitação, somente as tags de solicitação serão adicionadas ao snapshot. Se nenhuma tag for incluída na solicitação, as tags de cluster de cache serão adicionadas ao snapshot.

- Para snapshots automáticos:

As tags serão propagadas a partir das tags do grupo de replicação.

- **CreateServerlessCacheSnapshot :**

- Se as tags forem incluídas na solicitação, somente as tags de solicitação serão adicionadas ao snapshot de cache sem servidor.

- **CopySnapshot :**

- Se as tags forem incluídas na solicitação, somente as tags de solicitação serão adicionadas ao snapshot. Se nenhuma tag for incluída na solicitação, as tags de snapshot da origem serão adicionadas ao snapshot copiado.
- CopyServerlessCacheSnapshot :
 - Se as tags forem incluídas na solicitação, somente as tags de solicitação serão adicionadas ao snapshot de cache sem servidor.
- AddTagsToResource e RemoveTagsFromResource:
 - As tags serão added/removed do grupo de replicação e a ação será propagada para todos os clusters no grupo de replicação.

 Note

AddTagsToResource e RemoveTagsFromResource não pode ser usado para parâmetros e grupos de segurança padrão.

- IncreaseReplicaCount e ModifyReplicationGroupShardConfiguration:
 - Todos os novos clusters adicionados ao grupo de replicação terão as mesmas tags aplicadas como o grupo de replicação.

Restrições de tags

As restrições básicas a seguir se aplicam a tags:

- Número máximo de tags por recurso — 50
- Em todos os recursos, cada chave de tag deve ser exclusiva e possuir apenas um valor.
- Comprimento máximo da chave – 128 caracteres Unicode em UTF-8.
- Comprimento máximo do valor – 256 caracteres Unicode em UTF-8.
- Embora ElastiCache permita qualquer caractere em suas tags, outros serviços podem ser restritivos. Os caracteres permitidos nos serviços são: letras, números e espaços representáveis em UTF-8 e os seguintes caracteres: + - = . _ : / @
- As chaves e os valores de marcas diferenciam maiúsculas de minúsculas.
- O aws: prefixo está reservado para AWS uso. Não é possível editar nem excluir a chave ou o valor de uma tag quando ela tem uma chave de tag com esse prefixo. As tags com o prefixo aws: não contam para as tags por limite de recurso.

Você não pode encerrar, parar ou excluir um recurso baseado unicamente em suas tags; será preciso especificar o identificador de recursos. Por exemplo, para excluir snapshots marcados com uma chave de tag chamada DeleteMe, use a ação DeleteSnapshot com os identificadores de recursos dos snapshots, como snap-1234567890abcdef0.

Para obter mais informações sobre ElastiCache recursos que você pode marcar, consulte [Recursos que podem ser marcados](#).

Exemplo de marcação de recursos

- Criar um cache com tecnologia sem servidor usando tags. Este exemplo usa o Memcached como mecanismo.

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached \  
  --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Adicionar tags a um cache sem servidor

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Adição de tags a um grupo de replicação.

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-east-1:111111222233:replicationgroup:my-rg \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Criação de um cluster de cache usando tags.

```
aws elasticache create-cache-cluster \  
  --cluster-id testing-tags \  
  --cluster-description cluster-test \  
  --cache-subnet-group-name test \  
  --cache-node-type cache.t2.micro \  
  --engine valkey \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Criação de um cluster de cache usando tags. Este exemplo usa o Redis como mecanismo.


```
aws elasticache create-cache-cluster \  
--cluster-id testing-tags \  
--cluster-description cluster-test \  
--cache-subnet-group-name test \  
--cache-node-type cache.t2.micro \  
--engine valkey \  
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Criação de um snapshot sem servidor com tags. Este exemplo usa o Memcached como mecanismo.

```
aws elasticache create-serverless-cache-snapshot \  
--serverless-cache-name testing-tags \  
--serverless-cache-snapshot-name bkp-testing-tags-scs \  
--tags Key="work",Value="foo"
```

- Criação de um snapshot com tags.

No momento, os snapshots estão disponíveis apenas para Redis. Para esse caso, se você adicionar tags sob solicitação, mesmo que o grupo de replicação contenha tags, o snapshot receberá somente as tags da solicitação.

```
aws elasticache create-snapshot \  
--replication-group-id testing-tags \  
--snapshot-name bkp-testing-tags-rg \  
--tags Key="work",Value="foo"
```

Exemplos de políticas de controle de acesso baseadas em tags

1. Permissão de ação `AddTagsToResource` para um cluster somente se o cluster tiver a tag `Project=XYZ`.

JSON

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

```

        "Action": "elasticache:AddTagsToResource",
        "Resource": [
            "arn:aws:elasticache:*:*:cluster:*"
        ],
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/Project": "XYZ"
            }
        }
    }
]
}

```

2. Permissão de ação `RemoveTagsFromResource` de um grupo de replicação se ele contém as tags `Project` (Projeto) e `Service` (Serviço) e as chaves são diferentes de `Project` (Projeto) e `Service` (Serviço).

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticache:RemoveTagsFromResource",
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Service": "Elasticache",
                    "aws:ResourceTag/Project": "XYZ"
                },
                "ForAnyValue:StringNotEqualsIgnoreCase": {
                    "aws:TagKeys": [
                        "Project",
                        "Service"
                    ]
                }
            }
        }
    ]
}

```

3. Permissão de AddTagsToResource para qualquer recurso somente se as tags forem diferentes de Project (Projeto) e Service (Serviço).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:AddTagsToResource",
      "Resource": [
        "arn:aws:elasticache:*:*:*:*"
      ],
      "Condition": {
        "ForAnyValue:StringNotEqualsIgnoreCase": {
          "aws:TagKeys": [
            "Service",
            "Project"
          ]
        }
      }
    }
  ]
}
```

4. Negação da ação CreateReplicationGroup se a solicitação tiver Tag Project=Foo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CreateReplicationGroup",
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "Foo"
        }
      }
    }
  ]
}
```

```
    }
  ]
}
```

5. Negação da ação CopySnapshot se o snapshot de origem tiver a tag Project=XYZ e a tag de solicitação for Service=Elasticache.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CopySnapshot",
      "Resource": [
        "arn:aws:elasticache:*:*:snapshot:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "XYZ",
          "aws:RequestTag/Service": "Elasticache"
        }
      }
    }
  ]
}
```

6. Negar ação CreateCacheCluster se a tag de solicitação Project estiver faltando ou não for igual a Dev, QA ou Prod.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*",

```

```

        "arn:aws:elasticache:*:*:securitygroup:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    },
    {
        "Effect": "Deny",
        "Action": [
            "elasticache:CreateCacheCluster"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:cluster:*"
        ],
        "Condition": {
            "Null": {
                "aws:RequestTag/Project": "true"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateCacheCluster",
            "elasticache:AddTagsToResource"
        ],
        "Resource": "arn:aws:elasticache:*:*:cluster:*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/Project": [
                    "Dev",
                    "Prod",
                    "QA"
                ]
            }
        }
    }
]
}

```

Para obter informações relacionadas sobre chaves de condição, consulte [Uso de chaves de condição](#).

Monitoramento de custos com tags de alocação de custos

Ao adicionar etiquetas de alocação de custos aos seus recursos na Amazon ElastiCache, você pode rastrear os custos agrupando as despesas em suas faturas por valores de etiquetas de recursos.

Uma tag de alocação de ElastiCache custos é um par de valores-chave que você define e associa a um recurso. ElastiCache A chave e o valor diferenciam maiúsculas de minúsculas. Você pode usar uma chave de tag para definir uma categoria, e o valor da tag pode ser um item nessa categoria. Por exemplo, você pode definir uma chave de tag de `CostCenter` e um valor de tag de `10010`, indicando que o recurso está atribuído ao centro de custo 10010. Você também pode usar tags para designar recursos como sendo usados para teste ou produção, usando uma chave como `Environment` e valores como `test` ou `production`. Recomendamos que você use um conjunto consistente de chaves de tag para facilitar o rastreamento dos custos associados aos seus recursos.

Use etiquetas de alocação de custos para organizar sua AWS fatura de forma a refletir sua própria estrutura de custos. Para fazer isso, inscreva-se para receber a fatura AWS da sua conta com os valores-chave da tag incluídos. Então, para ver o custo de recursos combinados, organize suas informações de faturamento de acordo com recursos com os mesmos valores de chave de tags. Por exemplo, é possível marcar vários recursos com um nome de aplicação específico, e depois organizar suas informações de faturamento para ver o custo total daquela aplicação em vários serviços.

Você também pode combinar tags para rastrear custos com um maior nível de detalhes. Por exemplo, para rastrear seus custos de serviços por região, você pode usar as chaves de tag `Service` e `Region`. Em um recurso, você pode ter os valores `ElastiCache` e `Asia Pacific (Singapore)` e, em outro recurso, os valores `ElastiCache` e `Europe (Frankfurt)`. Em seguida, você pode ver seus ElastiCache custos totais divididos por região. Para saber mais, consulte [Usar tags de alocação de custos](#) no Guia do usuário do AWS Billing.

Você pode adicionar tags de alocação de ElastiCache custos a clusters baseados em ElastiCache nós. Ao adicionar, listar, modificar, copiar ou remover uma tag, a operação é aplicada somente ao cluster especificado.

Características das etiquetas de alocação de ElastiCache custos

- As tags de alocação de custos são aplicadas aos ElastiCache recursos especificados nas operações da CLI e da API como um ARN. O tipo de recurso será um "cluster".

ARN de exemplo: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- A chave de tags é o nome obrigatório da tag. O valor da string da chave pode ser de 1 a 128 caracteres Unicode e não pode ser prefixado com `aws:`. A string pode conter apenas o conjunto de letras Unicode, dígitos, espaços em branco, sublinhados (`_`), pontos finais (`.`), dois-pontos (`:`), barras invertidas (`\`), sinais de igualdade (`=`), sinais de adição (`+`), hífen (`-`) ou sinais de arroba (`@`).
- O valor da tag é o valor opcional da tag. O valor da string do valor pode ser de 1 a 256 caracteres Unicode e não pode ser prefixado com `aws:`. A string pode conter apenas o conjunto de letras Unicode, dígitos, espaços em branco, sublinhados (`_`), pontos finais (`.`), dois-pontos (`:`), barras invertidas (`\`), sinais de igualdade (`=`), sinais de adição (`+`), hífen (`-`) ou sinais de arroba (`@`).
- Um ElastiCache recurso pode ter no máximo 50 tags.
- Os valores não têm que ser exclusivos em um conjunto de tags. Por exemplo, você pode ter um conjunto de tags no qual as chaves `Service` e `Application` têm ambas o valor `ElastiCache`.

AWS não aplica nenhum significado semântico às suas tags. As tags são interpretadas estritamente como cadeias de caracteres. AWS não define automaticamente nenhuma tag em nenhum ElastiCache recurso.

Gerenciamento das suas tags de alocação de custos usando a AWS CLI

Você pode usar a AWS CLI para adicionar, modificar ou remover tags de alocação de custos.

Tags de alocação de custos são aplicadas em clusters do ElastiCache. O cluster que receberá tag é especificado usando um ARN (Nome de recurso da Amazon).

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Tópicos

- [Listagem de tags usando a AWS CLI](#)
- [Adição de tags usando a AWS CLI](#)
- [Modificação de tags usando a AWS CLI](#)

- [Remoção de tags usando a AWS CLI](#)

Listagem de tags usando a AWS CLI

Você pode usar a AWS CLI para listar tags em um recurso do ElastiCache existente usando a operação [list-tags-for-resource](#).

O código a seguir usa a AWS CLI para listar as tags no cluster Memcached `my-cluster` na região `us-west-2`.

Para Linux, macOS ou Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Para Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

O código a seguir usa a AWS CLI para listar as tags no nó `my-cluster-001` do Valkey ou Redis OSS no cluster `my-cluster` na região `us-west-2`.

Para Linux, macOS ou Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Para Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso.

```
{  
  "TagList": [  
    {  
      "Value": "10110",  
      "Key": "CostCenter"  
    },  
  ],  
}
```



```
{
  "Value": "EC2",
  "Key": "Service"
}
```

Se não houver tags no recurso, a saída será uma TagList vazia.

```
{
  "TagList": []
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [list-tags-for-resource](#).

Adição de tags usando a AWS CLI

Você pode usar a AWS CLI para adicionar tags a um recurso do ElastiCache existente usando a operação da CLI [add-tags-to-resource](#). Se a chave de tag não existir no recurso, a chave e o valor serão adicionados ao recurso. Se a chave já existir no recurso, o valor associado a essa chave será atualizado para o novo valor.

O código a seguir usa a AWS CLI para adicionar as chaves Service e Region com os valores elasticache e us-west-2, respectivamente, ao nó my-cluster-001 no cluster my-cluster na região us-west-2.

Memcached

Para 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
```

Para Windows:

```
aws elasticache add-tags-to-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
  --tags Key=Service,Value=elasticache ^
    Key=Region,Value=us-west-2
```

Redis do

Para Linux, macOS ou Unix:

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \  
  --tags Key=Service,Value=elasticache \  
         Key=Region,Value=us-west-2
```

Para Windows:

```
aws elasticache add-tags-to-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^  
  --tags Key=Service,Value=elasticache ^  
         Key=Region,Value=us-west-2
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso após a operação, conforme mostrado a seguir.

```
{  
  "TagList": [  
    {  
      "Value": "elasticache",  
      "Key": "Service"  
    },  
    {  
      "Value": "us-west-2",  
      "Key": "Region"  
    }  
  ]  
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [add-tags-to-resource](#).

Você também pode usar a AWS CLI para adicionar tags a um cluster ao criar um novo cluster usando a operação [create-cache-cluster](#). Não é possível adicionar tags ao criar um cluster usando o console de gerenciamento do ElastiCache. Depois que o cluster é criado, você pode usar o console para adicionar tags ao cluster.

Modificação de tags usando a AWS CLI

Você pode usar a AWS CLI para modificar as tags em um cluster do ElastiCache.

Para modificar tags:

- Use [add-tags-to-resource](#) para adicionar uma nova tag e um valor ou para alterar o valor associado a uma tag existente.
- Use [remove-tags-from-resource](#) para remover tags especificadas do recurso.

A saída de qualquer operação será uma lista de tags e seus valores no cluster especificado.

Remoção der tags usando a AWS CLI

Você pode usar a AWS CLI para remover tags de um cluster do ElastiCache for Memcached existente usando a operação [remove-tags-from-resource](#).

Para o Memcached, o código a seguir usa a AWS CLI para remover as etiquetas com as chaves Service e Region do nó my-cluster-001 no cluster my-cluster na região us-west-2.

Para 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
```

Para Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
  --tag-keys PM Service
```

Para o Redis OSS, o código a seguir usa a AWS CLI para remover as etiquetas com as chaves Service e Region do nó my-cluster-001 no cluster my-cluster na região us-west-2.

Para Linux, macOS ou Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \  
  --tag-keys PM Service
```

Para Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
```

```
--tag-keys PM Service
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso após a operação, conforme mostrado a seguir.

```
{  
  "TagList": []  
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [remove-tags-from-resource](#).

Gerenciamento das suas tags de alocação de custos usando a API do ElastiCache

Você pode usar a API do ElastiCache para adicionar, modificar ou remover tags de alocação de custos.

Tags de alocação de custos são aplicadas em clusters do ElastiCache for Memcached. O cluster que receberá tag é especificado usando um ARN (Nome de recurso da Amazon).

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Tópicos

- [Listagem de tags usando a API do ElastiCache](#)
- [Adição de tags usando a API do ElastiCache](#)
- [Modificação de tags usando a API do ElastiCache](#)
- [Remoção de tags usando a API do ElastiCache](#)

Listagem de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para listar tags em um recurso existente usando a operação [ListTagsForResource](#).

Para o Memcached, o código a seguir usa a API do ElastiCache para listar as tags no recurso `my-cluster` na região `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>
```

Para o Redis OSS, o código a seguir usa a API do ElastiCache para listar as tags no recurso `my-cluster-001` na região `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListTagsForResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Adição de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para adicionar tags a um cluster do ElastiCache existente usando a operação [AddTagsToResource](#). Se a chave de tag não existir no recurso, a chave e o valor serão adicionados ao recurso. Se a chave já existir no recurso, o valor associado a essa chave será atualizado para o novo valor.

O código a seguir usa a API do ElastiCache para adicionar as chaves `Service` e `Region` com os valores `elasticache` e `us-west-2` respectivamente. Para o Memcached, isso é aplicado ao recurso `my-cluster`. Para o Redis OSS, isso é aplicado ao recurso `my-cluster-001` na região `us-west-2`.

Memcached

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AddTagsToResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
```

```
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Redis do

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AddTagsToResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [AddTagsToResource](#) na Referência da API do Amazon ElastiCache.

Modificação de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para modificar as tags em um cluster do ElastiCache.

Para modificar o valor de uma tag:

- Use a operação [AddTagsToResource](#) para adicionar uma nova tag e um valor ou para alterar o valor de uma tag existente.
- Use [RemoveTagsFromResource](#) para remover tags do recurso.

A saída de qualquer operação será uma lista de tags e seus valores no recurso especificado.

Use [RemoveTagsFromResource](#) para remover tags do recurso.

Remoção de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para remover tags de um cluster do ElastiCache for Memcached existente usando a operação [RemoveTagsFromResource](#).

O código a seguir usa a API do ElastiCache para remover as tags com as chaves Service e Region do nó my-cluster-001 no cluster my-cluster na região us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RemoveTagsFromResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&TagKeys.member.1=Service  
&TagKeys.member.2=Region  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Usar a Lente do Well-Architected para o Amazon ElastiCache

Esta seção descreve o recurso Lente do Well-Architected para o Amazon ElastiCache, um conjunto de princípios de design e orientações para projetar workloads bem arquitetadas do ElastiCache.

- A Lente para o ElastiCache é um aditivo ao [AWS Well-Architected Framework](#).
- Cada pilar tem um conjunto de perguntas para ajudar a iniciar a discussão sobre uma arquitetura do ElastiCache.
 - Cada pergunta tem várias práticas principais, além de suas pontuações para fins de relatório.
 - Obrigatório: necessário antes de entrar em produção (a ausência representa alto risco).
 - Ideal: o melhor estado possível em que um cliente pode estar.
 - Bom: o que recomendamos para os clientes (a ausência representa risco médio).
- Terminologia do Well-Architected
 - [Componente](#): código, configuração e recursos da AWS que, juntos, atendem a um requisito. Os componentes interagem com outros componentes e geralmente são equivalentes a um serviço em arquiteturas de microsserviços.
 - [Workload](#): um conjunto de componentes que, juntos, agregam valor comercial. Exemplos de workloads são sites de marketing, sites de comércio eletrônico, back-ends de aplicações móveis, plataformas de análises etc.

Note

Este guia não foi atualizado para incluir informações sobre o cache com tecnologia sem servidor do ElastiCache e o novo mecanismo Valkey.

Tópicos

- [Pilar Excelência operacional da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar Segurança da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar Confiabilidade da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar de ElastiCache eficiência de desempenho da lente Amazon Well-Architected](#)
- [Pilar Otimização de custos da Lente do Well-Architected para o Amazon ElastiCache](#)

Pilar Excelência operacional da Lente do Well-Architected para o Amazon ElastiCache

O foco do pilar Excelência operacional está na execução e no monitoramento de sistemas para agregar valor empresarial e melhorar continuamente processos e procedimentos. Os principais tópicos incluem automatizar mudanças, responder a eventos e definir padrões para gerenciar operações diárias.

Tópicos

- [EO 1: Como você entende e responde aos alertas e eventos acionados pelo seu cluster do ElastiCache?](#)
- [EO 2: Quando e como você dimensiona seus clusters existentes do ElastiCache?](#)
- [EO 3: Como você gerencia os recursos do seu cluster do ElastiCache e mantém o cluster atualizado?](#)
- [EO 4: Como você gerencia as conexões de clientes aos seus clusters do ElastiCache?](#)
- [EO 5: Como você implanta componentes do ElastiCache para uma workload?](#)
- [EO 6: Como você planeja e mitiga falhas?](#)
- [EO 7: como solucionar problemas em eventos do mecanismo Valkey ou Redis OSS?](#)

EO 1: Como você entende e responde aos alertas e eventos acionados pelo seu cluster do ElastiCache?

Introdução: ao operar clusters do ElastiCache, você tem a opção de receber notificações e alertas quando eventos específicos ocorrerem. Por padrão, o ElastiCache registra em log [eventos](#) relacionados aos seus recursos, como um failover, substituição de nó, operação de ajuste de escala, manutenção programada e muito mais. Cada evento inclui a data e hora, o nome e tipo da origem e uma descrição.

Benefício: ser capaz de entender e gerenciar os motivos subjacentes aos eventos que acionam os alertas gerados pelo seu cluster permite que você opere com mais eficiência e responda aos eventos de forma adequada.

- [Obrigatório] Analise os eventos gerados pelo ElastiCache no console do ElastiCache (depois de selecionar sua região) ou usando o comando [describe-events](#) da [Amazon Command Line Interface](#) (AWS CLI) e a [API do ElastiCache](#). Configure o ElastiCache para enviar notificações sobre eventos de cluster importantes usando o Amazon Simple Notification Service (Amazon SNS). Usar o Amazon SNS com seus clusters permite que você realize ações programáticas em eventos do ElastiCache.
- Há duas grandes categorias de eventos: eventos atuais e programados. A lista de eventos atuais inclui: criação e exclusão de recursos, operações de ajuste de escala, failover, reinicialização de nó, criação de snapshot, modificação de parâmetros do cluster, renovação do certificado de CA, eventos de falha (falha no provisionamento do cluster [VPC ou ENI], falhas de ajuste de escala [ENI] e falhas de snapshot). A lista de eventos programados inclui: nó programado para substituição durante a janela de manutenção e substituição de nó reagendada.
- Embora talvez você não precise reagir imediatamente a alguns desses eventos, é fundamental examinar primeiro todos os eventos de falha:
 - ElastiCache:AddCacheNodeFailed
 - ElastiCache:CacheClusterProvisioningFailed
 - ElastiCache:CacheClusterScalingFailed
 - ElastiCache:CacheNodesRebooted
 - ElastiCache:SnapshotFailed (somente Valkey ou Redis OSS)
- [Recursos]:
 - [Gerenciando ElastiCache notificações do Amazon SNS](#)
 - [Notificações de eventos e o Amazon SNS](#)
- [Ideal] Para automatizar as respostas a eventos, utilize os recursos dos produtos e serviços da AWS, como o SNS e as funções do Lambda. Siga as práticas recomendadas ao fazer alterações pequenas, frequentes e reversíveis, como código para evoluir suas operações ao longo do tempo. Você deve usar as métricas do Amazon CloudWatch para monitorar seus clusters.

[Recursos]: [monitorar endpoints de réplica de leitura do ElastiCache \(modo cluster desabilitado\) usando o AWS Lambda, Amazon Route 53 e Amazon SNS](#) para um caso de uso do Lambda e SNS.

EO 2: Quando e como você dimensiona seus clusters existentes do ElastiCache?

Introdução: o dimensionamento correto do cluster do ElastiCache é um ato de equilíbrio que precisa ser avaliado sempre que houver alterações nos tipos de workload subjacentes. Seu objetivo é operar com o ambiente do tamanho certo para sua workload.

Benefício: a utilização excessiva de seus recursos pode resultar em latência elevada e diminuição geral da performance. Por outro lado, a subutilização pode resultar em provisionamento excessivo de recursos com uma otimização de custos não ideal. Ao dimensionar corretamente seus ambientes, você pode encontrar um equilíbrio entre eficiência de performance e otimização de custos. Para remediar o excesso ou a subutilização de recursos, o ElastiCache pode ter a escala ajustada em duas dimensões. Você pode ajustar a escala verticalmente ao aumentar ou diminuir a capacidade dos nós. Também pode ajustar a escala horizontalmente ao adicionar e remover nós.

- [Obrigatório] O excesso de utilização da CPU e da rede nos nós primários deve ser resolvido descarregando e redirecionando as operações de leitura para os nós de réplica. Use nós de réplica para operações de leitura a fim de reduzir a utilização do nó primário. Isso pode ser configurado em sua biblioteca de cliente do Valkey ou Redis OSS conectando-se ao endpoint leitor do ElastiCache para modo cluster desabilitado ou usando o comando READONLY para modo cluster habilitado.

[Recursos]:

- [Encontrar endpoints de conexão no ElastiCache](#)
- [Dimensionar um cluster corretamente](#)
- [Comando READONLY](#)
- [Obrigatório] Monitore a utilização de recursos críticos do cluster, como CPU, memória e rede. A utilização desses recursos específicos do cluster precisa ser monitorada para informar sua decisão de ajustar a escala e do tipo de operação de ajuste. Com o modo de cluster do ElastiCache desabilitado, o nó primário e os nós de réplica podem escalar verticalmente. Os nós de réplica também podem escalar horizontalmente, de 0 a 5 nós. Com o modo de cluster habilitado, o mesmo se aplica a cada fragmento do cluster. Além disso, você pode aumentar ou reduzir o número de fragmentos.

[Recursos]:

- [Monitorar as práticas recomendadas com o ElastiCache usando o Amazon CloudWatch](#)
- [Escalar clusters do ElastiCache para Valkey e Redis OSS](#)
- [Escalar clusters do ElastiCache para Memcached](#)

- [Ideal] O monitoramento da performance ao longo do tempo pode ajudar você a detectar mudanças na workload que passariam despercebidas se monitoradas em momentos específicos. Para detectar tendências de longo prazo, use as métricas do CloudWatch a fim de verificar intervalos de tempo mais longos. O aprendizado da observação de longos períodos de métricas do CloudWatch deve informar sua previsão sobre a utilização de recursos do cluster. Os pontos de dados e métricas do CloudWatch ficam disponíveis por até 455 dias.

[Recursos]:

- [Monitorar o ElastiCache com métricas do CloudWatch](#)
- [Monitorar o Memcached com métricas do CloudWatch](#)
- [Monitorar as práticas recomendadas com o ElastiCache usando o Amazon CloudWatch](#)
- [Ideal] Se seus recursos do ElastiCache forem criados com o CloudFormation, uma prática recomendada é realizar alterações usando modelos do CloudFormation para preservar a consistência operacional e evitar alterações de configuração não gerenciadas e desvios de pilha.

[Recursos]:

- [Referência de tipos de recurso do ElastiCache para o CloudFormation](#)
- [Ideal] Automatize suas operações de ajuste de escala usando dados operacionais de cluster e defina limites no CloudWatch para configurar alarmes. Use o CloudWatch Events e o Simple Notification Service (SNS) para acionar funções do Lambda e executar uma API do ElastiCache a fim de ajustar a escala dos clusters automaticamente. Um exemplo seria adicionar um fragmento ao cluster quando a métrica EngineCPUUtilization atingir 80% por um longo período. Outra opção seria usar DatabaseMemoryUsedPercentages para um limite baseado em memória.

[Recursos]:

- [Usar alarmes do Amazon CloudWatch](#)
- [O que são os eventos do Amazon CloudWatch?](#)
- [Como usar o AWS Lambda com o Amazon Simple Notification Service](#)
- [Referência da API do ElastiCache](#)

EO 3: Como você gerencia os recursos do seu cluster do ElastiCache e mantém o cluster atualizado?

Introdução: ao operar em grande escala, é essencial que você seja capaz de localizar e identificar todos os seus recursos do ElastiCache. Ao lançar novos atributos da aplicação, você precisa criar

simetria entre as versões do cluster em todos os tipos de ambiente do ElastiCache: desenvolvimento, teste e produção. Os atributos de recursos permitem que você separe ambientes para objetivos operacionais diferentes, como ao implantar novos atributos e ativar novos mecanismos de segurança.

Benefício: separar os ambientes de desenvolvimento, teste e produção é uma prática operacional recomendada. Também é uma prática recomendada que seus clusters e nós em todos os ambientes tenham os patches de software mais recentes aplicados usando processos bem compreendidos e documentados. Aproveitar os atributos nativos do ElastiCache permite que sua equipe de engenharia se concentre em cumprir os objetivos de negócios e não na manutenção do ElastiCache.

- [Ideal] Execute na versão mais recente do mecanismo disponível e aplique as atualizações de autoatendimento assim que estiverem disponíveis. O ElastiCache atualiza automaticamente sua infraestrutura subjacente durante a janela de manutenção especificada do cluster. No entanto, os nós em execução em seus clusters são atualizados por meio de atualizações de autoatendimento. Essas atualizações podem ser de dois tipos: patches de segurança ou pequenas atualizações de software. Compreenda a diferença entre os tipos de patches e quando eles são aplicados.

[Recursos]:

- [Atualizações de autoatendimento no Amazon ElastiCache](#)
- [Página de ajuda sobre manutenção gerenciada e atualizações de serviços do Amazon ElastiCache](#)
- [Ideal] Organize seus recursos do ElastiCache usando etiquetas. Use etiquetas em grupos de replicação e não em nós individuais. Você pode configurar etiquetas para serem exibidas ao consultar recursos e usar etiquetas para realizar pesquisas e aplicar filtros. Você deve usar grupos de recursos para criar e manter facilmente coleções de recursos que compartilham conjuntos comuns de etiquetas.

[Recursos]:

- [Práticas recomendadas de marcação](#)
- [Referência de tipos de recurso do ElastiCache para o CloudFormation](#)
- [Grupos de parâmetros](#)

EO 4: Como você gerencia as conexões de clientes aos seus clusters do ElastiCache?

Introdução: ao operar em grande escala, você precisa entender como seus clientes se conectam ao cluster do ElastiCache para gerenciar os aspectos operacionais da aplicação (como tempos de resposta).

Benefício: escolher o mecanismo de conexão mais adequado vai garantir que sua aplicação não se desconecte devido a erros de conectividade, como tempos limite.

- [Obrigatório] Separe as operações de leitura e gravação e conecte-se aos nós de réplica para executar as operações de leitura. No entanto, esteja ciente de que, ao separar as gravações das leituras, você perderá a capacidade de ler uma chave logo depois de gravá-la devido à natureza assíncrona da replicação do Valkey e do Redis OSS. O comando WAIT pode ser usado para aumentar a segurança dos dados no mundo real e forçar as réplicas a reconhecer as gravações antes de responder aos clientes, com um custo geral de performance. O uso de nós de réplica para operações de leitura pode ser configurado em sua biblioteca de cliente do ElastiCache usando o endpoint de leitor do ElastiCache para o modo de cluster desabilitado. Para habilitar o modo de cluster, use o comando READONLY. Para muitas das bibliotecas de cliente do ElastiCache, o READONLY é implementado por padrão ou por meio de uma configuração.

[Recursos]:

- [Encontrar endpoints de conexão no ElastiCache](#)
- [READONLY](#)
- [Obrigatório] Use o agrupamento de conexões. Estabelecer uma conexão TCP tem um custo de tempo de CPU nos lados do cliente e do servidor, e o agrupamento permite que você reutilize a conexão TCP.

Para reduzir a sobrecarga da conexão, use o agrupamento de conexões. Com um grupo de conexões, sua aplicação pode reutilizar e liberar conexões “à vontade”, sem o custo de estabelecer a conexão. Você pode implementar o agrupamento de conexões por meio de sua biblioteca de cliente do ElastiCache (se compatível), com um framework disponível para o ambiente da sua aplicação, ou criá-lo do zero.

- [Ideal] Certifique-se de que o tempo limite do soquete do cliente esteja definido para pelo menos um segundo (em vez do padrão típico de “nenhum” em vários clientes).
- Definir um valor de tempo limite muito baixo pode fazer com que o tempo limite seja atingido quando a carga do servidor estiver alta. Defini-lo muito alto pode fazer com que a aplicação demore muito para detectar problemas de conexão.

- Controle o volume de novas conexões implementando o agrupamento de conexões em sua aplicação de cliente. Isso reduz a latência e a utilização da CPU necessárias para abrir e fechar conexões e realizar um handshake de TLS, caso o TLS esteja habilitado no cluster.

[Recursos]: [configurar o ElastiCache para aumentar a disponibilidade](#)

- [Bom] Usar pipelines (quando seus casos de uso permitirem) pode aumentar significativamente a performance.
- Com pipelines, você reduz o tempo de ida e volta (RTT) entre os clientes da aplicação e o cluster, e novas solicitações podem ser processadas mesmo que o cliente ainda não tenha lido as respostas anteriores.
- Com pipelines, você pode enviar vários comandos para o servidor sem esperar por respostas ou confirmações. A desvantagem dos pipelines é que, quando você finalmente obtém todas as respostas em lote, pode ter ocorrido um erro que só será detectado no final.
- Implemente métodos para repetir as solicitações quando for retornado um erro que omite a solicitação incorreta.

[Recursos]: [Pipelines](#)

EO 5: Como você implanta componentes do ElastiCache para uma workload?

Introdução: os ambientes do ElastiCache podem ser implantados manualmente por meio do Console da AWS ou programaticamente por meio de APIs, CLI, kits de ferramentas etc. As práticas recomendadas de Excelência operacional sugerem que as implantações sejam automatizadas por meio de código sempre que possível. Além disso, os clusters do ElastiCache podem ser isolados por workload ou combinados para fins de otimização de custos.

Benefício: escolher o mecanismo de implantação mais adequado para seus ambientes do ElastiCache pode melhorar a excelência operacional ao longo do tempo. É recomendável realizar operações como código sempre que possível para minimizar a quantidade de erros humanos e aumentar a repetibilidade, a flexibilidade e o tempo de resposta aos eventos.

Ao entender os requisitos de isolamento da workload, você pode optar por ter ambientes dedicados do ElastiCache por workload ou combinar várias workloads em clusters únicos, ou combinações dessas estratégias. Compreender as vantagens e desvantagens pode ajudar a encontrar um equilíbrio entre Excelência operacional e Otimização de custos.

- [Obrigatório] Entenda as opções de implantação disponíveis para o ElastiCache e automatize esses procedimentos sempre que possível. Os possíveis caminhos de automação incluem CloudFormation, AWS CLI/SDK e APIs.

[Recursos]:

- [Referência de tipos de recurso do Amazon ElastiCache](#)
- [elasticache](#)
- [Referência da API do Amazon ElastiCache](#)
- [Obrigatório] Para todas as workloads, determine o nível de isolamento do cluster necessário.
 - [Ideal] Isolamento alto: mapeamento 1:1 entre workload e cluster. Permite o melhor controle sobre o acesso, o dimensionamento, a escalabilidade e o gerenciamento dos recursos do ElastiCache por workload.
 - [Melhor] Isolamento médio: M:1 isolado por finalidade, mas talvez compartilhado entre várias workloads (por exemplo, um cluster dedicado a armazenar workloads em cache e outro dedicado a mensagens).
 - [Bom] Isolamento baixo: M:1 multiuso, totalmente compartilhado. Recomendado para workloads em que o acesso compartilhado é aceitável.

EO 6: Como você planeja e mitiga falhas?

Introdução: a Excelência operacional inclui a antecipação de falhas por meio da realização de exercícios regulares “pré-mortem” a fim de identificar possíveis fontes de falha para que elas possam ser removidas ou mitigadas. O ElastiCache oferece uma API de failover que permite simular eventos de falha de nós, para fins de teste.

Benefício: ao testar cenários de falha com antecedência, você pode aprender como eles afetam sua workload. Isso permite testar com segurança os procedimentos de resposta e sua eficácia, além de familiarizar sua equipe com sua execução.

[Obrigatório] Realize regularmente testes de failover em contas de dev/teste. [TestFailover](#)

EO 7: como solucionar problemas em eventos do mecanismo Valkey ou Redis OSS?

Introdução: a Excelência operacional exige a capacidade de investigar as informações do serviço e do mecanismo para analisar a integridade e o status dos clusters. O ElastiCache pode emitir logs do mecanismo Valkey ou Redis OSS para o Amazon CloudWatch e o Amazon Kinesis Data Firehose.

Benefício: habilitar os logs do mecanismo Valkey ou Redis OSS nos clusters do ElastiCache fornece informações sobre eventos que afetam a integridade e desempenho dos clusters. Os logs do mecanismo Valkey ou Redis OSS fornecem dados diretamente do mecanismo que não estão disponíveis por meio do mecanismo de eventos do ElastiCache. Por meio da observação cuidadosa dos eventos do ElastiCache (consulte a pergunta EO 1 anterior) e dos logs do mecanismo, é possível determinar a ordem dos eventos ao solucionar problemas, tanto da perspectiva do serviço ElastiCache quanto da perspectiva do mecanismo.

- [Obrigatório] Certifique-se de que a funcionalidade de registro em log do mecanismo Redis OSS esteja habilitada. Ela está disponível no ElastiCache para Redis OSS versão 6.2 e posteriores. Isso pode ser feito durante a criação do cluster ou modificando o cluster depois da criação.
- Determine se o Amazon CloudWatch Logs ou o Amazon Kinesis Data Firehose são o destino apropriado para os logs do mecanismo Redis OSS.
- Selecione um log de destino apropriado no CloudWatch ou no Kinesis Data Firehose para manter os logs. Se você tiver vários clusters, considere usar um log de destino diferente para cada cluster, pois isso ajudará a isolar os dados ao solucionar problemas.

[Recursos]:

- Entrega de logs: [Entrega de logs](#)
- Destinos de registro em log: [Amazon CloudWatch Logs](#)
- Introdução ao Amazon CloudWatch Logs: [O que é o Amazon CloudWatch Logs?](#)
- Introdução ao Amazon Kinesis Data Firehose: [O que é o Amazon Kinesis Data Firehose?](#)
- [Ideal] Se estiver usando o Amazon CloudWatch Logs, considere usar o Amazon CloudWatch Logs Insights para consultar o log do mecanismo Valkey ou Redis OSS a fim de obter informações importantes.

Como exemplo, crie uma consulta ao grupo de logs do CloudWatch que contém os logs do mecanismo Valkey ou Redis OSS que retornarão eventos com um LogLevel de "WARNING", como:

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Recursos]: [Analisar dados de logs com o CloudWatch Logs Insights](#)

Pilar Segurança da Lente do Well-Architected para o Amazon ElastiCache

O foco do pilar Segurança está na proteção de informações e sistemas. Os principais tópicos incluem confidencialidade e integridade dos dados, identificação e gerenciamento de quem pode fazer o quê com o gerenciamento baseado em privilégios, proteção de sistemas e estabelecimento de controles para detectar eventos de segurança.

Tópicos

- [SEG 1: Quais etapas você está tomando para controlar o acesso autorizado aos dados do ElastiCache?](#)
- [SEG 2: Suas aplicações exigem autorização adicional para o ElastiCache além dos controles baseados em rede?](#)
- [SEG 3: Existe o risco de que os comandos possam ser executados acidentalmente, causando perda ou falha de dados?](#)
- [SEG 4: Como garantir a criptografia de dados em repouso com o ElastiCache?](#)
- [SEG 5: Como criptografar dados em trânsito com o ElastiCache?](#)
- [SEG 6: Como restringir o acesso aos recursos do ambiente de gerenciamento?](#)
- [SEG 7: Como detectar e responder a eventos de segurança?](#)

SEG 1: Quais etapas você está tomando para controlar o acesso autorizado aos dados do ElastiCache?

Introdução: todos os clusters do ElastiCache são projetados para serem acessados de instâncias do Amazon Elastic Compute Cloud em uma VPC, de funções sem servidor (AWS Lambda) ou de contêineres (Amazon Elastic Container Service). O cenário mais comum é acessar um cluster do ElastiCache de uma instância do Amazon Elastic Compute Cloud na mesma Amazon VPC (Amazon Virtual Private Cloud). Antes de poder se conectar a um cluster usando uma instância do Amazon EC2, você deve autorizar a instância do Amazon EC2 a acessar o cluster. Para acessar um cluster do ElastiCache em execução em uma VPC, é necessário conceder entrada de rede ao cluster.

Benefício: a entrada de rede no cluster é controlada por meio de grupos de segurança da VPC. Um grupo de segurança atua como firewall virtual para as instâncias do Amazon EC2 a fim de controlar o tráfego de entrada e saída. As regras de entrada controlam o tráfego de entrada para a instância e as regras de saída controlam o tráfego de saída da instância. No caso do ElastiCache, ao iniciar um cluster é necessário associar um grupo de segurança. Isso garante que as regras de tráfego de entrada e saída estejam em vigor para todos os nós que compõem o cluster. Além disso, o

ElastiCache está configurado para ser implantado exclusivamente em sub-redes privadas, de forma que elas só possam ser acessadas por meio de redes privadas da VPC.

- [Obrigatório] O grupo de segurança associado ao seu cluster controla a entrada na rede e o acesso ao cluster. Por padrão, um grupo de segurança não terá nenhuma regra de entrada definida e, portanto, nenhum caminho de entrada para o ElastiCache. Para habilitar isso, configure uma regra de entrada no grupo de segurança especificando o endereço/intervalo de IP de origem, o tráfego do tipo TCP e a porta do cluster do ElastiCache (porta padrão 6379 para o ElastiCache para Valkey e Redis OSS, por exemplo). Embora seja possível permitir um conjunto muito amplo de origens de entrada, como todos os recursos em uma VPC (0.0.0.0/0), é recomendável ser o mais granular possível ao definir as regras de entrada, como autorizar somente o acesso de entrada a clientes Valkey ou Redis OSS em execução em instâncias do Amazon EC2 associadas a um grupo de segurança específico.

[Recursos]:

- [Sub-redes e grupos de sub-redes](#)
- [Acessar um cluster ou grupo de replicação](#)
- [Controlar o tráfego para recursos usando grupos de segurança](#)
- [Grupos de segurança do Amazon Elastic Compute Cloud para instâncias do Linux](#)
- [Obrigatório] As políticas do AWS Identity and Access Management podem ser atribuídas a funções do AWS Lambda que permitam acessar os dados do ElastiCache. Para habilitar esse atributo, crie uma função de execução do IAM com a permissão `AWSLambdaVPCAccessExecutionRole` e atribua o perfil à função AWS Lambda.

[Recursos]: Configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC: [Tutorial: Configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC](#)

SEG 2: Suas aplicações exigem autorização adicional para o ElastiCache além dos controles baseados em rede?

Introdução: em cenários em que é necessário restringir ou controlar o acesso aos clusters individualmente para cada cliente, é recomendável fazer a autenticação por meio do comando AUTH. Os tokens de autenticação do ElastiCache, com gerenciamento opcional de usuários e grupos de usuários, possibilitam que o ElastiCache exija uma senha antes de permitir que os clientes executem comandos e chaves de acesso, aumentando assim a segurança do plano de dados.

Benefício: para ajudar a manter os dados em segurança, o ElastiCache fornece mecanismos de proteção contra o acesso não autorizado aos dados. Isso inclui impor que o AUTH do controle de acesso baseado em perfil (RBAC) ou um token de AUTH (senha) sejam usados pelos clientes para que se conectem ao ElastiCache antes de executar comandos autorizados.

- [Ideal] No ElastiCache versão 6.x e superior para Redis OSS, e ElastiCache versão 7.2 e superior para Valkey, defina os controles de autenticação e autorização definindo grupos de usuários, usuários e strings de acesso. Atribua usuários a grupos de usuários, depois atribua grupos de usuários a clusters. Para utilizar o RBAC, ele deve ser selecionado na criação do cluster e a criptografia em trânsito deve estar habilitada. Use um cliente Valkey ou Redis OSS compatível com TLS para poder aproveitar o RBAC.

[Recursos]:

- [Aplicação do RBAC a um grupo de replicação para o ElastiCache](#)
- [Especificação de permissões usando uma string de acesso](#)
- [ACL](#)
- [Versões aceitas do ElastiCache](#)
- [Ideal] Em versões do ElastiCache anteriores à 6.x para Redis OSS, além de definir token/senha fortes e manter uma política de senha rígida para AUTH, a prática recomendada é alternar a senha/token. O ElastiCache pode gerenciar até dois (2) tokens de autenticação por vez. Você também pode modificar o cluster para exigir explicitamente o uso de tokens de autenticação.

[Recursos]: [Modificação do token AUTH em um cluster existente do ElastiCache](#)

SEG 3: Existe o risco de que os comandos possam ser executados acidentalmente, causando perda ou falha de dados?

Introdução: há vários comandos do Valkey ou Redis OSS que podem ter impactos adversos nas operações se executados por engano ou por agentes mal-intencionados. Esses comandos podem ter consequências não intencionais do ponto de vista da performance e da segurança dos dados. Por exemplo, um desenvolvedor pode chamar rotineiramente o comando FLUSHALL em um ambiente de desenvolvimento e, devido a um erro, pode tentar inadvertidamente chamar esse comando em um sistema de produção, resultando em perda acidental de dados.

Benefício: a partir do ElastiCache versão 5.0.3 para Redis OSS, é possível renomear certos comandos que podem atrapalhar sua workload. Renomear os comandos pode ajudar a evitar que sejam executados acidentalmente no cluster.

- [Obrigatório]

[Recursos]:

- [ElastiCache versão 5.0.3 para Redis OSS \(obsoleto, use a versão 5.0.6\)](#)
- [ElastiCache versão 5.0.3 para alterações de parâmetros do Redis OSS](#)
- [Segurança do Redis OSS](#)

SEG 4: Como garantir a criptografia de dados em repouso com o ElastiCache?

Introdução: embora o ElastiCache seja um datastore em memória, é possível criptografar qualquer dado que possa ser persistido (no armazenamento) como parte das operações padrão do cluster. Isso inclui backups programados e manuais gravados no Amazon S3, bem como dados salvos no armazenamento em disco como resultado de operações de sincronização e troca. Os tipos de instância nas famílias M6g e R6g também oferecem criptografia sempre ativa em memória.

Benefício: o ElastiCache fornece criptografia em repouso opcional para aumentar a segurança dos dados.

- [Obrigatório] A criptografia em repouso só pode ser habilitada em um cluster do ElastiCache (grupo de replicação) durante sua criação. Um cluster existente não pode ser modificado para começar a criptografar dados em repouso. Por padrão, o ElastiCache fornecerá e gerenciará as chaves usadas na criptografia em repouso.

[Recursos]:

- [Restrições da criptografia em repouso](#)
- [Ativar criptografia em repouso](#)
- [Ideal] Utilize os tipos de instância do Amazon EC2 que criptografam dados enquanto eles estão na memória (como M6g ou R6g). Sempre que possível, considere gerenciar suas próprias chaves para criptografia em repouso. Para ambientes de segurança de dados mais rigorosos, o AWS Key Management Service (KMS) pode ser usado para autogerenciar as chaves mestras de cliente (CMK). Por meio da integração do ElastiCache com o AWS Key Management Service, você pode criar, possuir e gerenciar as chaves usadas para criptografia de dados em repouso em seu cluster do ElastiCache.

[Recursos]:

- [Uso de chaves gerenciadas pelo cliente do AWS Key Management Service](#)
- [AWS Key Management Service](#)

- [Conceitos do AWS KMS](#)

SEG 5: Como criptografar dados em trânsito com o ElastiCache?

Introdução: é um requisito comum evitar que os dados sejam comprometidos em trânsito. Isso representa dados dentro de componentes de um sistema distribuído, bem como entre clientes de aplicações e nós de cluster. O ElastiCache oferece suporte a esse requisito ao permitir a criptografia de dados em trânsito entre clientes e cluster e entre os próprios nós do cluster. Os tipos de instância nas famílias M6g e R6g também oferecem criptografia sempre ativa em memória.

Benefício: a criptografia em trânsito do Amazon ElastiCache é um atributo opcional que permite aumentar a segurança de seus dados nos pontos mais vulneráveis: quando estão em trânsito de um local para outro.

- [Obrigatório] A criptografia em trânsito só pode ser habilitada em um cluster (grupo de replicação) durante sua criação. Observe que, devido ao processamento adicional necessário para criptografar/descriptografar dados, a implementação da criptografia em trânsito vai afetar a performance. Para entender o impacto, é recomendável comparar a workload antes e depois de ativar a criptografia em trânsito.

[Recursos]:

- [Visão geral da criptografia em trânsito](#)

SEG 6: Como restringir o acesso aos recursos do ambiente de gerenciamento?

Introdução: as políticas do IAM e o ARN habilitam controles de acesso refinados no ElastiCache para Valkey e Redis OSS, permitindo um controle mais rígido para gerenciar a criação, modificação e exclusão de clusters do ElastiCache para Redis.

Benefício: o gerenciamento dos recursos do Amazon ElastiCache, como grupos de replicação, nós etc., pode ser restrito a contas da AWS que têm permissões específicas com base nas políticas do IAM, aumentando a segurança e a confiabilidade dos recursos.

- [Obrigatório] Gerencie o acesso aos recursos do Amazon ElastiCache atribuindo políticas do AWS Identity and Access Management específicas aos usuários da AWS, viabilizando um controle mais preciso sobre quais contas podem realizar quais ações nos clusters.

[Recursos]:

- [Visão geral do gerenciamento de permissões de acesso aos recursos do ElastiCache](#)
- [Uso de políticas baseadas em identidade \(políticas do IAM\) para o Amazon ElastiCache](#)

SEG 7: Como detectar e responder a eventos de segurança?

Introdução: quando implantado com o RBAC habilitado, o ElastiCache exporta métricas do CloudWatch para notificar os usuários sobre eventos de segurança. Essas métricas ajudam a identificar tentativas fracassadas de autenticação, acesso a chaves ou execução de comandos para os quais os usuários do RBAC não têm autorização.

Além disso, os recursos de produtos e serviços da AWS ajudam a proteger a workload geral automatizando as implantações e registrando em log todas as ações e modificações para revisão/ auditoria posterior.

Benefício: ao monitorar eventos, sua organização consegue responder de acordo com seus requisitos, políticas e procedimentos. Automatizar o monitoramento e as respostas a esses eventos de segurança fortalece sua postura geral de segurança.

- [Obrigatório] Familiarize-se com as métricas publicadas do CloudWatch relacionadas às falhas de autenticação e autorização do RBAC.
 - AuthenticationFailures = tentativas fracassadas de autenticação no Valkey ou Redis OSS
 - KeyAuthorizationFailures = tentativas fracassadas de usuários para acessar chaves sem permissão
 - CommandAuthorizationFailures = tentativas fracassadas de usuários para executar comandos sem permissão

[Recursos]:

- [Métricas para o Valkey ou Redis OSS](#)
- [Ideal] É recomendável configurar alertas e notificações sobre essas métricas e responder conforme necessário.

[Recursos]:

- [Usar alarmes do Amazon CloudWatch](#)
- [Ideal] Use o comando ACL LOG do Valkey ou Redis OSS para obter mais detalhes

[Recursos]:

- [ACL LOG](#)

- [Ideal] Familiarize-se com os recursos de produtos e serviços da AWS relacionados ao monitoramento, registro e análise de implantações e eventos do ElastiCache.

[Recursos]:

- [Logs das chamadas de API do Amazon ElastiCache com o AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-backup-check](#)
- [Monitorar o uso com métricas do CloudWatch](#)

Pilar Confiabilidade da Lente do Well-Architected para o Amazon ElastiCache

O pilar de confiabilidade se concentra nas workloads que executam as funções pretendidas e em como se recuperar rapidamente de falhas para atender às demandas. Os principais tópicos incluem projeto de sistema distribuído, planejamento de recuperação e adaptação às mudanças de requisitos.

Tópicos

- [CONF 1: Como você está oferecendo suporte a implantações de arquitetura de alta disponibilidade \(HA\)?](#)
- [CONF 2: Como você está cumprindo seus objetivos de ponto de recuperação \(RPOs\) com o ElastiCache?](#)
- [CONF 3: Como você oferece suporte aos requisitos de recuperação de desastres \(DR\)?](#)
- [CONF 4: Como se planejar efetivamente para os failovers?](#)
- [CONF 5: Seus componentes do ElastiCache foram projetados para escalar?](#)

CONF 1: Como você está oferecendo suporte a implantações de arquitetura de alta disponibilidade (HA)?

Introdução: compreender a arquitetura de alta disponibilidade do Amazon ElastiCache permitirá que você opere em um estado resiliente durante eventos de disponibilidade.

Benefício: arquitetar seus clusters do ElastiCache para que sejam resilientes a falhas garante maior disponibilidade para suas implantações do ElastiCache.

- [Obrigatório] Determine o nível de confiabilidade necessário para seu cluster do ElastiCache. Workloads diferentes têm padrões de resiliência diferentes, desde workloads totalmente efêmeras

até workloads essenciais à missão. Defina as necessidades de cada tipo de ambiente que você opera, como desenvolvimento, teste e produção.

Mecanismo de armazenamento em cache: ElastiCache para Memcached versus ElastiCache para Valkey e Redis OSS

1. O ElastiCache para Memcached não fornece nenhum mecanismo de replicação e é usado principalmente para workloads efêmeras.
 2. O ElastiCache para Redis OSS oferece os atributos de HA discutidos abaixo.
- [Ideal] Para workloads que exigem HA, use o ElastiCache no modo de cluster com no mínimo duas réplicas por fragmento, mesmo para workloads com requisitos de throughput baixa que exijam apenas um fragmento.

1. Com o modo de cluster habilitado, o multi-AZ é habilitado automaticamente.

O multi-AZ minimiza o tempo de inatividade realizando failovers automáticos do nó primário para as réplicas, em caso de manutenção planejada ou não planejada, além de mitigar falhas em AZ.

2. Para workloads fragmentadas, uma quantidade mínima de três fragmentos fornece uma recuperação mais rápida durante eventos de failover, pois o protocolo de cluster do Valkey ou Redis OSS exige que a maioria dos nós primários esteja disponível para ter quórum.
3. Configure duas ou mais réplicas em toda a disponibilidade.

Ter duas réplicas proporciona maior escalabilidade de leitura e também disponibilidade de leitura em cenários em que uma réplica passa por manutenção.

4. Use tipos de nó baseados em Graviton2 (nós padrão na maioria das regiões).

O ElastiCache apresenta desempenho otimizado nesses nós. Como resultado, você obtém melhor performance de replicação e sincronização, resultando em maior disponibilidade geral.

5. Monitore e ajuste o tamanho certo para lidar com picos de tráfego previstos: sob carga pesada, o mecanismo pode deixar de responder, o que afeta a disponibilidade. BytesUsedForCache e DatabaseMemoryUsagePercentage são bons indicadores do uso da memória, enquanto ReplicationLag é um indicador da integridade de replicação com base na taxa de gravação. Você pode usar essas métricas para acionar o ajuste de escala do cluster.
6. Garanta a resiliência do lado do cliente testando com a [API de failover antes de um evento de failover na produção](#).

[Recursos]:

- [Configurar o ElastiCache para Redis OSS para aumentar a disponibilidade](#)
- [Alta disponibilidade com o uso de grupos de replicação](#)

CONF 2: Como você está cumprindo seus objetivos de ponto de recuperação (RPOs) com o ElastiCache?

Introdução: entenda o RPO da workload para embasar as decisões sobre estratégias de backup e recuperação do ElastiCache.

Benefício: ter uma estratégia de RPO em vigor pode melhorar a continuidade dos negócios nos cenários de recuperação de desastres. Projetar políticas de backup e restauração pode ajudar você a atingir seus objetivos de ponto de recuperação (RPO) para seus dados do ElastiCache. O ElastiCache oferece recursos de snapshot que são armazenados no Amazon S3, com uma política de retenção configurável. Esses instantâneos são gerados durante uma janela de backup definida e gerenciados automaticamente pelo serviço. Se sua workload exigir granularidade de backup adicional, você tem a opção de criar até 20 backups manuais por dia. Os backups criados manualmente não têm uma política de retenção de serviços e podem ser mantidos indefinidamente.

- [Obrigatório] Entenda e documente o RPO de suas implantações do ElastiCache.
 - Lembre-se de que o Memcached não oferece nenhum processo de backup.
 - Analise a capacidade dos atributos de backup e restauração do ElastiCache.
- [Ideal] Implemente um processo bem comunicado para fazer backup do cluster.
 - Inicie backups manuais conforme necessário.
 - Analise as políticas de retenção para backups automáticos.
 - Observe que os backups manuais serão mantidos indefinidamente.
 - Agende seus backups automáticos durante períodos de baixo uso.
 - Execute operações de backup em réplicas de leitura para garantir a minimização do impacto na performance do cluster.
- [Bom] Aproveite o atributo de backup agendado do ElastiCache para fazer backup regular de seus dados durante uma janela definida.
 - Teste periodicamente as restaurações de seus backups.
- [Recursos]:
 - [Redis OSS](#)
 - [Backup e restauração do ElastiCache](#)

- [Criação de backups manuais](#)
- [Programação de backups automáticos](#)
- [Backup e restauração de clusters do ElastiCache](#)

CONF 3: Como você oferece suporte aos requisitos de recuperação de desastres (DR)?

Introdução: a recuperação de desastres é um aspecto importante do planejamento de qualquer workload. O ElastiCache oferece várias opções para implementar a recuperação de desastres com base nos requisitos de resiliência da workload. Com o Amazon ElastiCache Global Datastore, você pode gravar em seu cluster em uma região e disponibilizar os dados para leitura em outros dois clusters de réplica entre regiões, viabilizando leituras de baixa latência e recuperação de desastres em várias regiões.

Benefício: compreender e se planejar para uma variedade de cenários de desastre pode garantir a continuidade dos negócios. As estratégias de DR devem equilibrar custo, impacto na performance e potencial de perda de dados.

- [Obrigatório] Desenvolva e documente estratégias de DR para todos os seus componentes do ElastiCache com base nos requisitos da workload. O ElastiCache é único porque alguns casos de uso são totalmente efêmeros e não exigem nenhuma estratégia de DR, enquanto outros estão na extremidade oposta do espectro e exigem uma estratégia de DR extremamente robusta. Todas as opções devem ser ponderadas em relação à otimização de custos: maior resiliência requer mais recursos de infraestrutura.

Entenda as opções de DR disponíveis em nível regional e multirregional.

- As implantações multi-AZ são recomendadas para evitar falhas de AZ. Realize a implantação com o modo de cluster habilitado em arquiteturas multi-AZ, com um mínimo de 3 AZs disponíveis.
- O Global Datastore é recomendado para se proteger contra falhas regionais.
- [Ideal] Habilite o Global Datastore para workloads que exigem resiliência por região.
 - Tenha um plano para realizar failover para a região secundária em caso de degradação da primária.
 - Teste o processo de failover multirregional antes de um failover na produção.
 - Monitore a métrica `ReplicationLag` para entender o impacto potencial da perda de dados durante eventos de failover.

- [Recursos]:
 - [Atenuar falhas](#)
 - [Replicação entre regiões da AWS usando datastores globais](#)
 - [Restauração de um backup com redimensionamento opcional do cluster](#)
 - [Minimizar o tempo de inatividade no ElastiCache para Valkey e Redis OSS usando o Multi-AZ](#)

CONF 4: Como se planejar efetivamente para os failovers?

Introdução: habilitar o multi-AZ com failovers automáticos é uma prática recomendada do ElastiCache. Em certos casos, o ElastiCache para Valkey e Redis OSS substitui os nós primários como parte das operações de serviço. Exemplos incluem eventos de manutenção planejada e o caso improvável de falha em um nó ou problema em zona de disponibilidade. Os failovers bem-sucedidos dependem da configuração do ElastiCache e da sua biblioteca de cliente.

Benefício: seguir as práticas recomendadas para failovers do ElastiCache em conjunto com sua biblioteca de cliente específica do ElastiCache ajuda a minimizar o tempo de inatividade potencial durante eventos de failover.

- [Obrigatório] Com o modo de cluster desabilitado, use tempos limite para que seus clientes detectem se precisam se desconectar do nó primário antigo e se reconectar ao novo nó primário, usando o endereço IP do endpoint primário atualizado. Com o modo de cluster habilitado, a biblioteca de cliente é responsável por detectar alterações na topologia subjacente do cluster. Isso é feito com mais frequência por meio de configurações na biblioteca de cliente do ElastiCache, que também permitem definir a frequência e o método de atualização. Cada biblioteca de cliente oferece configurações próprias e mais detalhes estão disponíveis na documentação correspondente.

[Recursos]:

- [Minimizar o tempo de inatividade no ElastiCache para Valkey e Redis OSS usando o Multi-AZ](#)
- Analise as práticas recomendadas da sua biblioteca de cliente do ElastiCache.
- [Obrigatório] Os failovers bem-sucedidos dependem de um ambiente de replicação saudável entre o nó primário e os nós de réplica. Analise e compreenda a natureza assíncrona da replicação do Valkey e Redis OSS, bem como as métricas do CloudWatch disponíveis para relatar o atraso na replicação entre o nó primário e os nós de réplica. Para casos de uso que exigem maior segurança de dados, use o comando WAIT para forçar as réplicas a reconhecerem as gravações antes de responder aos clientes conectados.

[Recursos]:

- [Métricas para o Valkey ou Redis OSS](#)
- [Monitorar as práticas recomendadas com o ElastiCache usando o Amazon CloudWatch](#)
- [Ideal] Valide regularmente a capacidade de resposta da aplicação durante o failover usando a API Test Failover do ElastiCache.

[Recursos]:

- [Testar o failover automático para uma réplica de leitura no ElastiCache](#)
- [Teste do failover automático](#)

CONF 5: Seus componentes do ElastiCache foram projetados para escalar?

Introdução: quando você entende os recursos de ajuste de escala e as topologias de implantação disponíveis, seus componentes do ElastiCache podem se ajustar ao longo do tempo para atender às mudanças nos requisitos da workload. O ElastiCache oferece ajuste de escala em quatro direções: aumento/redução na horizontal e na vertical.

Benefício: seguir as práticas recomendadas para implantações do ElastiCache oferece a maior flexibilidade para ajuste de escala, além de atender ao princípio do Well Architected de ajuste horizontal da escala para minimizar o impacto das falhas.

- [Obrigatório] Entenda a diferença entre topologias com modo de cluster habilitado e desabilitado. Em quase todos os casos, é recomendável realizar a implantação com o modo de cluster habilitado, pois isso aumenta a escalabilidade ao longo do tempo. Os componentes com modo de cluster desabilitado têm capacidade limitada de escalar horizontalmente com a adição de réplicas de leitura.
- [Obrigatório] Entenda quando e como escalar.
 - Para mais READIOPS: adicione réplicas.
 - Para mais WRITEOPS: adicione fragmentos (aumentar a escala horizontalmente).
 - Para mais E/S de rede: use instâncias otimizadas para rede (aumentar a escala verticalmente).
- [Ideal] Implante seus componentes do ElastiCache com o modo de cluster habilitado, com uma tendência para mais nós menores, em vez de menos nós maiores. Isso limita o raio de alcance de uma falha de nó.
- [Ideal] Inclua réplicas em seus clusters para melhorar a capacidade de resposta durante eventos de ajuste de escala.

- [Bom] Com o modo de cluster desabilitado, utilize as réplicas de leitura para aumentar a capacidade geral de leitura. O ElastiCache oferece suporte a até 5 réplicas de leitura com o modo de cluster desabilitado, além de ajuste vertical da escala.
- [Recursos]:
 - [Escalar clusters do ElastiCache](#)
 - [Aumento de escala vertical on-line](#)

Pilar de ElastiCache eficiência de desempenho da lente Amazon Well-Architected

O pilar Eficiência de performance enfoca o uso eficiente dos recursos de TI e computação. Os principais tópicos incluem a seleção dos tipos e tamanhos certos de recursos com base nos requisitos da workload, o monitoramento da performance e a tomada de decisões informadas para manter a eficiência à medida que as necessidades dos negócios evoluem.

Tópicos

- [PE 1: Como você monitora o desempenho do seu ElastiCache cluster Amazon?](#)
- [PE 2: Como você está distribuindo o trabalho entre os nós ElastiCache do cluster?](#)
- [EP 3: Para workloads de armazenamento em cache, como você monitora e relata a eficácia e a performance do cache?](#)
- [EP 4: Como sua workload otimiza o uso de recursos e conexões de rede?](#)
- [PE 5: Como você gerencia o and/or despejo por exclusão de chaves?](#)
- [PE 6: Como você modela e interage com os dados ElastiCache?](#)
- [PE 7: Como você registra comandos de execução lenta no seu ElastiCache cluster da Amazon?](#)
- [PE8: Como o Auto Scaling ajuda a aumentar o desempenho do ElastiCache cluster?](#)

PE 1: Como você monitora o desempenho do seu ElastiCache cluster Amazon?

Introdução: ao entender as métricas de monitoramento existentes, você pode identificar a utilização atual. O monitoramento adequado pode ajudar a identificar possíveis gargalos que afetam a performance de um cluster.

Benefício: a compreensão das métricas associadas ao seu cluster pode ajudar a orientar técnicas de otimização que podem levar à redução da latência e ao aumento do throughput.

- [Obrigatório] Teste de referência da performance usando um subconjunto da workload.
 - Você deve monitorar a performance da workload real usando mecanismos, como testes de carga.
 - Monitore as CloudWatch métricas durante a execução desses testes para obter uma compreensão das métricas disponíveis e estabelecer uma linha de base de desempenho.
- [Melhor] ElastiCache Para cargas de trabalho do Valkey e do Redis OSS, renomeie comandos computacionalmente caros KEYS, como para limitar a capacidade dos usuários de executar comandos de bloqueio em clusters de produção.
 - ElastiCache cargas de trabalho que executam o engine 6.x para Redis OSS podem aproveitar o controle de acesso baseado em funções para restringir determinados comandos. O acesso aos comandos pode ser controlado criando usuários e grupos de usuários com o AWS console ou a CLI e associando os grupos de usuários a um cluster. No Redis OSS 6, quando o RBAC está habilitado, podemos usar “-@dangerous” para não permitir que determinado usuário execute comandos caros, como KEYS, MONITOR, SORT etc.
 - Para o mecanismo versão 5.x, renomeie os comandos usando o parâmetro `rename-commands` no grupo de parâmetros do cluster.
- [Melhor] Analise consultas lentas e procure técnicas de otimização.
 - ElastiCache Para cargas de trabalho do Valkey e do Redis OSS, saiba mais sobre suas consultas analisando o Slow Log. Por exemplo, você pode usar o comando `valkey-cli slowlog get 10` para mostrar os últimos 10 comandos que excederam o limite de latência (10 milissegundos por padrão).
 - Certas consultas podem ser realizadas com mais eficiência usando estruturas de dados complexas ElastiCache para Valkey e Redis OSS. Como exemplo, para pesquisas de intervalo de estilo numérico, uma aplicação pode implementar índices numéricos simples com conjuntos ordenados. O gerenciamento desses índices pode reduzir as verificações realizadas no conjunto de dados e retornar dados com maior eficiência de performance.
 - ElastiCache Para cargas de trabalho Valkey e Redis OSS, `redis-benchmark` fornece uma interface simples para testar o desempenho de diferentes comandos usando entradas definidas pelo usuário, como número de clientes e tamanho dos dados.
 - Como o Memcached só oferece suporte a comandos simples em nível de chave, considere criar chaves adicionais como índices para evitar a iteração no espaço de chaves a fim de atender às consultas do cliente.
- [Recursos]:
 - [Monitorando o uso com CloudWatch métricas](#)

- [Usando CloudWatch alarmes da Amazon](#)
- [Parâmetros específicos do Valkey e do Redis OSS](#)
- [SLOWLOG](#)
- [comparativo](#)

PE 2: Como você está distribuindo o trabalho entre os nós ElastiCache do cluster?

Introdução em nível de pergunta: A forma como seu aplicativo se conecta aos ElastiCache nós da Amazon pode afetar o desempenho e a escalabilidade do cluster.

Benefício: o uso adequado dos nós disponíveis no cluster garantirá que o trabalho seja distribuído entre os recursos disponíveis. As técnicas a seguir também ajudam a evitar recursos ociosos.

- [Obrigatório] Faça com que os clientes se conectem ao ElastiCache endpoint adequado.
 - ElastiCache para Valkey e Redis, o OSS implementa endpoints diferentes com base no modo de cluster em uso. Para o modo de cluster ativado, ElastiCache fornecerá um endpoint de configuração. Para o modo de cluster desativado, ElastiCache fornece um endpoint primário, normalmente usado para gravações, e um endpoint de leitura para balancear leituras entre réplicas. A implementação correta desses endpoints resultará em melhor performance e operações de escalabilidade mais fáceis. Evite conectar-se a endpoints de nós individuais, a menos que haja um requisito específico que justifique isso.
 - Para clusters Memcached de vários nós, ElastiCache fornece um endpoint de configuração que permite a descoberta automática. É recomendável usar um algoritmo de hash para distribuir o trabalho uniformemente entre os nós de cache. Muitas bibliotecas de clientes do Memcached implementam hash consistente. Verifique a documentação da biblioteca que você está usando para ver se ela oferece suporte para hashing consistente e como implementá-lo. Você pode encontrar mais informações sobre a implementação desses atributos [aqui](#).
- [Melhor] Aproveite os clusters habilitados ElastiCache para o modo de cluster Valkey e Redis OSS para melhorar a escalabilidade.
 - ElastiCache para Valkey e Redis OSS (modo de cluster ativado), os clusters oferecem suporte a [operações de escalabilidade on-line](#) (out/in and up/down) para ajudar a distribuir dados dinamicamente entre fragmentos. O uso do endpoint de configuração vai garantir que os clientes com reconhecimento de cluster possam se ajustar às mudanças na topologia do cluster.

- Você também pode reequilibrar o cluster movendo os hashslots entre os fragmentos disponíveis em seu cluster ElastiCache for Valkey e Redis OSS (modo de cluster ativado). Isso ajuda a distribuir o trabalho de modo mais eficiente entre os fragmentos disponíveis.
- [Melhor] Implemente uma estratégia para identificar e corrigir chaves “hot” na workload.
- Considere o impacto das estruturas de dados multidimensionais do Valkey ou Redis OSS, como listas, fluxos, conjuntos etc. Essas estruturas de dados são armazenadas em chaves únicas, que residem em um único nó. Uma chave multidimensional muito grande tem o potencial de utilizar mais capacidade de rede e memória do que outros tipos de dados e pode provocar um uso desproporcional desse nó. Se possível, projete sua workload para distribuir o acesso aos dados entre várias chaves distintas.
- As chaves “hot” na workload podem afetar a performance do nó em uso. ElastiCache Para cargas de trabalho Valkey e Redis OSS, você pode detectar teclas de atalho usando `valkey-cli --hotkeys` se uma política de memória máxima de LFU estiver em vigor.
- Considere replicar chaves “hot” entre vários nós para distribuir o acesso a eles de forma mais uniforme. Essa abordagem exige que o cliente grave em vários nós primários (o próprio nó do Valkey ou Redis OSS não fornecerá essa funcionalidade) e mantenha uma lista de nomes de chaves para leitura, além do nome da chave original.
- ElastiCache [O motor 7.2 para Valkey e superior e a ElastiCache versão 6 para Redis OSS e superior oferecem suporte ao cache do lado do cliente assistido pelo servidor](#). Isso permite que os aplicativos aguardem alterações em uma chave antes de fazer chamadas de rede de volta para ElastiCache.
- [Recursos]:
 - [Configure ElastiCache para Valkey e Redis OSS para maior disponibilidade](#)
 - [Encontrar endpoints de conexão no ElastiCache](#)
 - [Práticas recomendadas para balanceamento de carga](#)
 - [Refragmentação online para o Valkey ou o Redis OSS \(modo cluster habilitado\)](#)
 - [Armazenamento em cache do lado do cliente no Valkey e Redis OSS](#)

EP 3: Para workloads de armazenamento em cache, como você monitora e relata a eficácia e a performance do cache?

Introdução em nível de pergunta: O armazenamento em cache é uma carga de trabalho comum ElastiCache e é importante que você entenda como gerenciar a eficácia e o desempenho do cache.

Benefício: sua aplicação pode mostrar sinais de performance lenta. Sua capacidade de usar métricas específicas de cache para informar sua decisão sobre como aumentar a performance da aplicação é essencial para sua workload de cache.

- [Obrigatório] Meça e acompanhe a taxa de acertos de cache ao longo do tempo. A eficiência do seu cache é determinada pela “taxa de acertos de cache”. A taxa de acertos de cache é definida pelo total de acertos de chave dividido pelo total de acertos e erros. Quanto mais próxima de 1 for a taxa, mais eficaz será o cache. Uma taxa de acertos de cache baixa é decorrente do volume de erros de cache. Os erros de cache ocorrem quando a chave solicitada não é encontrada no cache. Uma chave não está no cache porque ela foi removida ou excluída, expirou ou nunca existiu. Entenda por que as chaves não estão no cache e desenvolva estratégias apropriadas para incluí-las no cache.

[Recursos]:

- [Métricas para o Valkey e Redis OSS](#)
- [Obrigatório] Meça e colete o desempenho do cache do aplicativo em conjunto com os valores de latência e utilização da CPU para entender se você precisa fazer ajustes no seu time-to-live ou em outros componentes do aplicativo. ElastiCache fornece um conjunto de CloudWatch métricas para latências agregadas para cada estrutura de dados. Essas métricas de latência são calculadas usando a estatística commandstats do comando INFO e não incluem o tempo de rede e E/S. Esse é apenas o tempo consumido ElastiCache para processar as operações.

[Recursos]:

- [Métricas para o Valkey e Redis OSS](#)
- [Monitorando as melhores práticas com ElastiCache o uso da Amazon CloudWatch](#)
- [Ideal] Escolha a estratégia de cache certa para suas necessidades. Uma taxa de acertos de cache baixa é decorrente do volume de erros de cache. Se sua workload foi projetada para ter um baixo volume de erros de cache (como comunicação em tempo real), é melhor realizar análises de suas estratégias de armazenamento em cache e aplicar as resoluções mais apropriadas para sua workload, como instrumentação de consulta para medir a memória e a performance. As estratégias implementadas para preencher e manter seu cache dependem de quais dados seus clientes precisam armazenar em cache e dos padrões de acesso a esses dados. Por exemplo, é improvável que você use a mesma estratégia para recomendações personalizadas em uma aplicação de streaming e para notícias em alta.

[Recursos]:

- [Estratégias de armazenamento em cache para Memcached](#)

- [Práticas recomendadas de armazenamento em cache](#)
- [Desempenho em grande escala com o ElastiCache Whitepaper da Amazon](#)

EP 4: Como sua workload otimiza o uso de recursos e conexões de rede?

Introdução em nível de pergunta: ElastiCache para Valkey, Memcached e Redis, o OSS é suportado por muitos clientes de aplicativos, e as implementações podem variar. Você precisa entender o gerenciamento de redes e conexões em vigor para analisar o impacto potencial na performance.

Benefício: o uso eficiente dos recursos de rede pode melhorar a eficiência da performance do seu cluster. As recomendações a seguir podem reduzir as demandas de rede e melhorar a latência e o throughput do cluster.

- [Obrigatório] Gerencie proativamente as conexões com seu ElastiCache cluster.
 - O agrupamento de conexões na aplicação reduz a sobrecarga criada no cluster devido à abertura e ao encerramento de conexões. Monitore o comportamento da conexão na Amazon CloudWatch usando `CurrConnections` `NewConnections` e.
 - Evite a fuga de conexões ao encerrar adequadamente as conexões de clientes, quando apropriado. As estratégias de gerenciamento de conexões incluem o encerramento adequado das conexões que não estão em uso e a definição de tempo limite para conexões.
 - Para workloads do Memcached, há uma quantidade configurável de memória reservada para lidar com conexões chamada `memcached_connections_overhead`.
- [Melhor] Compacte objetos grandes para reduzir a memória e melhorar o throughput da rede.
 - A compactação de dados pode reduzir a quantidade necessária de throughput de rede (Gbps), mas aumenta a quantidade de trabalho na aplicação para compactar e descompactar dados.
 - A compactação também reduz a quantidade de memória consumida pelas chaves.
 - Com base nas necessidades da sua aplicação, considere as diferenças entre taxa de compressão e velocidade de compressão.
- [Recursos]:
 - [ElastiCache - Armazenamento de dados global](#)
 - [Parâmetros específicos do Memcached](#)
 - [ElastiCache a versão 5.0.3 para Redis OSS aprimora o manuseio para aumentar o desempenho I/O](#)
 - [Métricas para o Valkey e Redis OSS](#)

- [Configure ElastiCache para maior disponibilidade](#)

PE 5: Como você gerencia o and/or despejo por exclusão de chaves?

Introdução em nível de pergunta: as cargas de trabalho têm requisitos e comportamentos esperados diferentes quando um nó de cluster está se aproximando dos limites de consumo de memória.

ElastiCache tem políticas diferentes para lidar com essas situações.

Benefício: o gerenciamento adequado da memória disponível e a compreensão das políticas de remoção vão ajudar a garantir o conhecimento do comportamento do cluster quando os limites de memória da instância forem excedidos.

- [Obrigatório] Instrumente o acesso aos dados para avaliar qual política aplicar. Identifique uma política de memória máxima apropriada para controlar se e como as remoções são realizadas no cluster.
 - A remoção ocorre quando a memória máxima do cluster é consumida e há uma política em vigor para permitir a remoção. O comportamento do cluster nessa situação depende da política de remoção especificada. Essa política pode ser gerenciada usando `maxmemory-policy` no grupo de parâmetros de cluster.
 - A política padrão `volatile-lru` libera memória ao remover chaves com um prazo de validade definido (valor de TTL). As políticas menos usadas (LFU) e menos usadas recentemente (LRU) removem as chaves com base no uso.
 - Para workloads do Memcached, existe uma política LRU padrão que controla as remoções em cada nó. O número de despejos em seu ElastiCache cluster da Amazon pode ser monitorado usando a métrica de despejos na Amazon. CloudWatch
- [Melhor] Padronize o comportamento de exclusão para controlar o impacto na performance de seu cluster e evitar gargalos de performance inesperados.
 - ElastiCache Para cargas de trabalho do Valkey e do Redis OSS, ao remover explicitamente as chaves do cluster, `UNLINK` é como `DEL`: ele remove as chaves especificadas. No entanto, o comando executa a recuperação real da memória em um thread diferente, portanto não bloqueio, enquanto `DEL` bloqueia. A remoção real ocorrerá posteriormente de forma assíncrona.
 - Para a ElastiCache versão 6.x para cargas de trabalho do Redis OSS, o comportamento do `DEL` comando pode ser modificado no grupo de parâmetros usando o parâmetro. `lazyfree-lazy-user-del`
- [Recursos]:

- [Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache](#)
- [UNLINK](#)
- [Gerenciamento financeiro em nuvem com AWS](#)

PE 6: Como você modela e interage com os dados ElastiCache?

Introdução em nível de pergunta: ElastiCache depende muito do aplicativo das estruturas de dados e do modelo de dados usados, mas também precisa considerar o armazenamento de dados subjacente (se presente). Entenda as estruturas de dados disponíveis e garanta que você esteja usando as estruturas de dados mais adequadas às suas necessidades.

Benefício em nível de pergunta: a modelagem de dados ElastiCache tem várias camadas, incluindo casos de uso de aplicativos, tipos de dados e relacionamentos entre elementos de dados. Além disso, cada tipo de dado e comando tem suas próprias assinaturas de performance bem documentadas.

- [Ideal] Uma das práticas recomendadas é reduzir a substituição não intencional de dados. Use uma convenção de nomenclatura que minimize a sobreposição de nomes de chave. A nomenclatura convencional de suas estruturas de dados usa um método hierárquico, como `APPNAME:CONTEXT:ID` e `ORDER-APP:CUSTOMER:123`.

[Recursos]:

- [Nomenclatura de chaves](#)
- [Melhor] ElastiCache para Valkey e Redis, os comandos OSS têm uma complexidade de tempo definida pela notação Big O. Desta vez, a complexidade de um comando é uma *algorithmic/mathematical* representação de seu impacto. Ao introduzir um novo tipo de dado em sua aplicação, você precisa analisar cuidadosamente a complexidade temporal dos comandos relacionados. Os comandos com uma complexidade temporal de $O(1)$ são constantes no tempo e não dependem do tamanho da entrada, enquanto os comandos com uma complexidade temporal de $O(N)$ são lineares no tempo e estão sujeitos ao tamanho da entrada. Devido ao design de thread único do Valkey e do ElastiCache Redis OSS, um grande volume de operações de alta complexidade de tempo resultará em menor desempenho e em possíveis tempos limite de operação.

[Recursos]:

- [Comandos](#)

- [Melhor] Use APIs para obter visibilidade da GUI no modelo de dados em seu cluster.

[Recursos]:

- [Comandante do Redis OSS](#)
- [Navegador do Redis OSS](#)
- [Redsmin](#)

PE 7: Como você registra comandos de execução lenta no seu ElastiCache cluster da Amazon?

Introdução: benefícios do ajuste de performance por meio da captura, agregação e notificação de comandos de longa execução. Ao entender quanto tempo leva para que os comandos sejam executados, você pode determinar quais comandos resultam em baixo desempenho, bem como os comandos que impedem o desempenho ideal do mecanismo. ElastiCache também tem a capacidade de encaminhar essas informações para a Amazon CloudWatch ou o Amazon Kinesis Data Firehose.

Benefício: manter um log em um local permanente dedicado e fornecer eventos de notificação para comandos lentos podem ajudar na análise detalhada da performance e podem servir para acionar eventos automatizados.

- [Obrigatório] ElastiCache executando um mecanismo Valkey 7.2 ou mais recente, ou executando um mecanismo Redis OSS versão 6.0 ou mais recente, grupo de parâmetros configurado corretamente e registro do SLOWLOG ativado no cluster.
 - Os parâmetros necessários só estarão disponíveis quando a compatibilidade da versão do mecanismo estiver definida como Valkey 7.2 e posterior, ou Redis OSS versão 6.0 ou posterior.
 - O registro em log SLOWLOG ocorre quando o tempo de execução de um comando pelo servidor é maior que um valor especificado. O comportamento do cluster depende dos parâmetros do grupo de parâmetros associado, que são `slowlog-log-slower-than` e `slowlog-max-len`.
 - As alterações terão efeito imediatamente.
- [Melhor] Aproveite nossos recursos do CloudWatch Kinesis Data Firehose.
 - Use os recursos de filtragem e alarme do CloudWatch CloudWatch Logs Insights e do Amazon Simple Notification Services para obter monitoramento de desempenho e notificação de eventos.
 - Use os recursos de streaming do Kinesis Data Firehose para arquivar logs SLOWLOG no armazenamento permanente ou para acionar o ajuste automático de parâmetros do cluster.
 - Determine qual formato atende melhor às suas necessidades, JSON ou texto simples.

- Forneça permissões do IAM para publicar no CloudWatch ou no Kinesis Data Firehose.
- [Melhor] Configure `slowlog-log-slower-than` com um valor diferente do padrão.
 - Esse parâmetro determina por quanto tempo um comando pode ser executado no mecanismo Valkey ou Redis OSS antes de ser registrado em log como um comando lento. O valor padrão é 10.000 microssegundos (10 milissegundos). O valor padrão pode ser muito alto para algumas workloads.
 - Determine um valor que seja mais adequado para sua workload com base nas necessidades da aplicação e nos resultados dos testes, mas lembre-se de que um valor muito baixo pode gerar dados em excesso.
- [Melhor] Mantenha o valor padrão de `slowlog-max-len`.
 - Esse parâmetro determina o limite superior de quantos comandos lentos são capturados na memória do Valkey ou Redis OSS por vez. O valor 0 desabilita a captura. Quanto maior o valor, mais entradas serão armazenadas na memória, reduzindo a chance de informações importantes serem removidas antes que possam ser revisadas. O valor padrão é 128.
 - O valor padrão é adequado para a maioria das workloads. Se houver necessidade de analisar dados em uma janela de tempo expandida a partir de `valkey-cli` por meio do comando `SLOWLOG`, considere aumentar esse valor. Isso permite que mais comandos permaneçam na memória do Valkey ou Redis OSS.

Se você estiver emitindo os dados do `SLOWLOG` para o CloudWatch Logs ou para o Kinesis Data Firehose, os dados persistirão e poderão ser analisados fora do sistema, reduzindo ElastiCache a necessidade de armazenar um grande número de comandos de execução lenta na memória Valkey ou Redis OSS.

- [Recursos]:
 - [Como ativo o log lento em um cluster?](#)
 - [Entrega de logs](#)
 - [Parâmetros específicos do Redis OSS](#)
 - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
 - [Amazon Kinesis Data Firehose](#)

PE8: Como o Auto Scaling ajuda a aumentar o desempenho do ElastiCache cluster?

Introdução em nível de pergunta: ao implementar o recurso de escalonamento automático Valkey ou Redis OSS, ElastiCache seus componentes podem se ajustar com o tempo para aumentar ou

diminuir automaticamente os fragmentos ou réplicas desejados. Isso pode ser feito implementando a política de rastreamento de metas ou de ajuste de escala programado.

Benefício em nível de pergunta: compreender e planejar os picos na carga de trabalho pode garantir um desempenho aprimorado do armazenamento em cache e a continuidade dos negócios. ElastiCache O Auto Scaling monitora continuamente a utilização da CPU/memória para garantir que o cluster esteja operando nos níveis de desempenho desejados.

- [Obrigatório] Ao iniciar um cluster ElastiCache para Valkey ou Redis OSS:
 1. Verifique se o modo de cluster está habilitado.
 2. Verifique se a instância pertence a uma família de determinado tipo e tamanho compatíveis com o ajuste de escala automático.
 3. Verifique se o cluster não está sendo executado em datastores globais, Outposts ou zonas locais.

[Recursos]:

- [Escalabilidade de clusters no Valkey e Redis OSS \(modo cluster habilitado\)](#)
- [Usar o ajuste de escala automático com fragmentos](#)
- [Usar o ajuste de escala automático com réplicas](#)
- [Ideal] Identifique se sua workload exige muita leitura ou gravação para definir a política de ajuste de escala. Para obter a melhor performance, use apenas uma métrica de rastreamento. É recomendável evitar várias políticas para cada dimensão, pois as políticas de ajuste de escala automático aumentam a escala horizontalmente quando a meta é atingida, mas só aumentam a escala verticalmente quando todas as políticas de rastreamento de metas estiverem prontas para aumentar a escala verticalmente.

[Recursos]:

- [Políticas de ajuste de escala automático](#)
- [Definir uma política de ajuste de escala](#)
- [Ideal] O monitoramento da performance ao longo do tempo pode ajudar você a detectar mudanças na workload que passariam despercebidas se monitoradas em momentos específicos. Você pode analisar as CloudWatch métricas correspondentes para a utilização do cluster em um período de quatro semanas para determinar o limite do valor-alvo. Se você ainda não tiver certeza de qual valor escolher, recomendamos começar com o valor mínimo de métrica predefinido compatível.

[Recursos]:

- [Monitorando o uso com CloudWatch métricas](#)
- [Melhor] Recomendamos testar seu aplicativo com as cargas de trabalho mínimas e máximas esperadas, para identificar o número exato de cargas de trabalho shards/replicas necessárias para que o cluster desenvolva políticas de escalabilidade e mitigue problemas de disponibilidade.

[Recursos]:

- [Registrar uma meta escalável](#)
- [Registrando um alvo escalável usando o AWS CLI](#)

Pilar Otimização de custos da Lente do Well-Architected para o Amazon ElastiCache

O foco do pilar Otimização de custos é evitar custos desnecessários. Os principais tópicos incluem entender e controlar onde o dinheiro está sendo gasto, selecionar o tipo de nó mais adequado (usar instâncias que compatíveis com a hierarquização de dados com base nas necessidades da workload), a quantidade certa de tipos de recurso (quantidade de réplicas de leitura), a análise de gastos ao longo do tempo e a escalabilidade para atender às necessidades da empresa sem gastos excessivos.

Tópicos

- [CUSTOS 1: Como você identifica e monitora os custos associados aos seus recursos do ElastiCache? Como você desenvolve mecanismos para permitir que os usuários criem, gerenciem e descartem os recursos criados?](#)
- [CUSTOS 2: Como você usa ferramentas de monitoramento contínuo para ajudar a otimizar os custos associados aos seus recursos do ElastiCache?](#)
- [CUSTOS 3: Você deve usar um tipo de instância que ofereça suporte à hierarquização de dados? Quais são as vantagens de uma hierarquização de dados? Quando não usar instâncias de hierarquização de dados?](#)

CUSTOS 1: Como você identifica e monitora os custos associados aos seus recursos do ElastiCache? Como você desenvolve mecanismos para permitir que os usuários criem, gerenciem e descartem os recursos criados?

Introdução: entender as métricas de custo requer a participação e a colaboração de várias equipes: engenharia de software, gerenciamento de dados, proprietários de produtos, finanças e liderança. A identificação dos principais fatores de custo exige que todas as partes envolvidas compreendam as alavancas de controle do uso do serviço e as vantagens e desvantagens do gerenciamento de custos, e essa costuma ser a principal diferença entre ações bem-sucedidas e não tão bem-sucedidas de otimização de custos. Garantir que você tenha processos e ferramentas em vigor para monitorar os recursos criados desde o desenvolvimento até a produção e a retirada ajuda a gerenciar os custos associados ao ElastiCache.

Benefício: o acompanhamento contínuo de todos os custos associados à workload exige uma compreensão profunda da arquitetura que inclui o ElastiCache como um de seus componentes. Além disso, você deve ter um plano de gerenciamento de custos para coletar e comparar o uso com seu orçamento.

- [Obrigatório] Institua um Centro de Excelência da Nuvem (CCoE) com um de seus estatutos fundadores para definir, monitorar e agir com base nas métricas relacionadas ao uso do ElastiCache em suas organizações. Se existir um CCoE funcional, garanta que ele saiba como ler e acompanhar os custos associados ao ElastiCache. Ao criar recursos, use políticas e perfis do IAM para validar que somente equipes e grupos específicos possam instanciar recursos. Isso vai garantir que os custos estejam associados aos resultados comerciais e que uma linha clara de responsabilidade seja estabelecida, do ponto de vista dos custos.
 1. O CCoE deve identificar, definir e publicar métricas de custo que são atualizadas mensalmente sobre o uso principal do ElastiCache em dados categóricos, como:
 - a. Tipos de nó usados e seus atributos: instâncias padrão versus instâncias otimizadas para memória, instâncias sob demanda versus instâncias reservadas, regiões e zonas de disponibilidade
 - b. Tipos de ambiente: gratuito, desenvolvimento, teste e produção
 - c. Estratégias de backup, armazenamento e retenção
 - d. Transferência de dados dentro de uma região e entre regiões
 - e. Instâncias em execução no Amazon Outposts

2. O CCoE consiste em uma equipe multifuncional com representação não exclusiva das equipes de engenharia de software, gerenciamento de dados, produtos, finanças e liderança em sua organização.

[Recursos]:

- [Criar um Centro de Excelência da Nuvem](#)
 - [Preço do Amazon ElastiCache](#)
- [Obrigatório] Use etiquetas de alocação de custos para monitorar os custos em um baixo nível de granularidade. Use o Gerenciamento de Custos da AWS para visualizar, entender e gerenciar seus custos e uso da AWS ao longo do tempo.
 1. Use etiquetas para organizar seus recursos e etiquetas de alocação de custos para monitorar os custos da AWS em detalhes. Depois de ativar as etiquetas de alocação de custos, a AWS as utiliza para organizar seus custos de recursos no relatório de alocação de custos para facilitar a categorização e o controle dos seus custos na AWS. A AWS fornece dois tipos de etiquetas de alocação de custos: etiquetas geradas pela AWS e etiquetas definidas por usuário. A AWS define, cria e aplica as etiquetas geradas pela AWS para você, enquanto você define, cria e aplica as etiquetas definidas por usuário. É necessário ativar os dois tipos de etiquetas separadamente para que elas possam ser exibidas no Gerenciamento de Custos ou em um relatório de alocação de custos.
 2. Use tags de alocação de custos para organizar sua fatura da AWS para refletir sua própria estrutura de custos. Quando você adiciona etiquetas de alocação de custos aos seus recursos no Amazon ElastiCache, pode acompanhar os custos agrupando as despesas nas suas faturas por valores de etiqueta de recursos. Considere combinar etiquetas para monitorar os custos com um maior nível de detalhes.

[Recursos]:

- [Usar etiquetas de alocação de custos da AWS](#)
 - [Monitorar os custos com etiquetas de alocação de custos](#)
 - [Explorador de Custos da AWS](#)
- [Ideal] Conecte o custo do ElastiCache às métricas que abrangem toda a organização.
 1. Considere as métricas de negócios e as métricas operacionais, como latência: quais conceitos em seu modelo de negócios são compreensíveis em todas as funções? As métricas precisam ser compreensíveis pelo maior número possível de funções na organização.

2. Exemplos: usuários atendidos simultaneamente, latência máxima e média por operação e usuário, pontuações de engajamento do usuário, taxas de retorno do usuário/semana, duração da sessão/usuário, taxa de abandono, taxa de acertos de cache e chaves monitoradas.

[Recursos]:

- [Monitorar o uso com métricas do CloudWatch](#)
- [Bom] Mantenha uma visibilidade arquitetônica e operacional atualizada sobre métricas e custos em toda a workload que usa o ElastiCache.
 1. Entenda todo o seu ecossistema de soluções. O ElastiCache tende a fazer parte de um ecossistema completo de serviços da AWS em seu conjunto de tecnologias, desde clientes até API Gateway, Redshift e QuickSight para ferramentas de geração de relatórios (por exemplo).
 2. Mapeie os componentes da sua solução, desde clientes, conexões, segurança, operações em memória, armazenamento, automação de recursos, acesso e gerenciamento de dados, em seu diagrama de arquitetura. Cada camada se conecta a toda a solução e tem suas próprias necessidades e recursos que aumentam e/ou ajudam a gerenciar o custo geral.
 3. Seu diagrama deve incluir o uso de políticas de computação, rede, armazenamento, ciclo de vida, coleta de métricas, bem como os elementos operacionais e funcionais do ElastiCache da sua aplicação.
 4. É provável que os requisitos de uma workload evoluam com o tempo e é essencial que você continue a manter e documentar sua compreensão dos componentes subjacentes, bem como de seus principais objetivos funcionais, a fim de manter a proatividade no gerenciamento de custos da workload.
 5. O suporte executivo para visibilidade, responsabilização, priorização e recursos é crucial para que você tenha uma estratégia eficaz de gerenciamento de custos para o ElastiCache.

CUSTOS 2: Como você usa ferramentas de monitoramento contínuo para ajudar a otimizar os custos associados aos seus recursos do ElastiCache?

Introdução: você precisa buscar um equilíbrio adequado entre os custos do ElastiCache e as métricas de performance da aplicação. O Amazon CloudWatch fornece visibilidade das principais métricas operacionais que podem te ajudar a avaliar se seus recursos do ElastiCache estão sendo usados em excesso ou subutilizados em relação às suas necessidades. Do ponto de vista da otimização de custos, você precisa entender quando há excesso de provisionamento e ser capaz de desenvolver mecanismos apropriados para redimensionar seus recursos do ElastiCache, ao mesmo tempo em que mantém suas necessidades de operação, disponibilidade, resiliência e performance.

Benefício: em um estado ideal, você terá provisionado recursos suficientes para atender às necessidades operacionais de sua workload e não terá recursos subutilizados que possam levar a um estado de custo abaixo do ideal. Você precisa ser capaz de identificar e evitar a operação de recursos superdimensionados do ElastiCache por longos períodos.

- [Obrigatório] Use o CloudWatch para monitorar seus clusters do ElastiCache e analisar como essas métricas se relacionam com seus painéis do Explorador de custos da AWS.
 1. O ElastiCache fornece métricas em nível de host (por exemplo, uso da CPU) e métricas que são específicas do software do mecanismo de cache (por exemplo, obtenções e ausências no cache). Essas métricas são medidas e publicadas para cada nó de cache em intervalos de 60 segundos.
 2. As métricas de performance do ElastiCache (CPUUtilization, EngineUtilization, SwapUsage, CurrConnections e Evictions) podem indicar que você precisa aumentar/diminuir a escala verticalmente (usar tipos de nó de cache maiores/menores) ou aumentar/diminuir a escala horizontalmente (adicionar mais/menos fragmentos). Entenda as implicações de custo das decisões de ajuste de escala criando uma matriz de playbook que estima o custo adicional e os períodos mínimo e máximo necessários para atingir os limites de performance de sua aplicação.

[Recursos]:

- [Monitorar o uso com métricas do CloudWatch](#)
- [Quais métricas devo monitorar?](#)
- [Preço do Amazon ElastiCache](#)
- [Obrigatório] Entenda e documente sua estratégia de backup e as implicações de custo.
 1. Com o ElastiCache, os backups são armazenados no Amazon S3, que fornece armazenamento durável. Você precisa entender as implicações de custo em relação à sua capacidade de se recuperar de falhas.
 2. Ative backups automáticos que vão excluir os arquivos de backup que ultrapassaram o limite de retenção.

[Recursos]:

- [Programação de backups automáticos](#)
- [Preços do Amazon Simple Storage Service](#)
- [Ideal] Use nós reservados para suas instâncias como uma estratégia deliberada para gerenciar custos de workloads que sejam bem compreendidas e documentadas. Para nós reservados, é necessário pagar uma taxa adiantada que depende do tipo de nó e da duração da reserva, um

ou três anos. Essa cobrança é muito menor do que a cobrança de uso por hora dos nós sob demanda.

1. Talvez seja necessário operar os clusters do ElastiCache usando nós sob demanda até reunir dados suficientes para estimar os requisitos da instância reservada. Planeje e documente os recursos necessários para atender às suas necessidades e compare os custos esperados em todos os tipos de instância (sob demanda versus reservada)
2. Avalie regularmente os novos tipos de nó de cache disponíveis e avalie se faz sentido, do ponto de vista das métricas de operação e custo, migrar sua frota de instâncias para novos tipos de nó de cache.

CUSTOS 3: Você deve usar um tipo de instância que ofereça suporte à hierarquização de dados? Quais são as vantagens de uma hierarquização de dados? Quando não usar instâncias de hierarquização de dados?

Introdução: selecionar o tipo de instância apropriado pode afetar não só a performance e o serviço, mas também as finanças. Cada tipo de instância tem custos diferentes. Selecionar um ou alguns tipos de instâncias grandes que possam acomodar todas as necessidades de armazenamento em memória pode ser uma decisão natural. No entanto, isso pode afetar significativamente os custos à medida que o projeto amadurece. Garantir que o tipo de instância correto seja selecionado exige uma avaliação periódica do tempo de inatividade do objeto do ElastiCache.

Benefício: você deve ter uma compreensão clara de como os vários tipos de instância afetam seus custos no presente e no futuro. Mudanças marginais ou periódicas na workload não devem provocar mudanças desproporcionais nos custos. Se a workload permitir, os tipos de instância compatíveis com a hierarquização de dados oferecem um preço melhor por armazenamento disponível. Devido à disponibilidade por instância de armazenamento SSD, as instâncias com hierarquização de dados oferecem suporte a uma capacidade total de dados muito maior por instância.

- [Obrigatório] Entenda as limitações das instâncias com hierarquização de dados.
 1. Disponível somente para clusters do ElastiCache para Valkey ou Redis OSS.
 2. Somente alguns tipos de instância oferecem suporte à hierarquização de dados.
 3. Somente o ElastiCache versão 6.2 para Redis e superior é compatível.
 4. Itens grandes não são trocados para SSD. Objetos com mais de 128 MiB são mantidos na memória.

[Recursos]:

- [Hierarquização de dados](#)
- [Preço do Amazon ElastiCache](#)
- [Obrigatório] Entenda qual porcentagem do seu banco de dados é acessada regularmente pela workload.
 1. As instâncias com hierarquização de dados são ideais para workloads que geralmente acessam uma pequena parte do conjunto geral de dados, mas que ainda exigem acesso rápido aos dados restantes. Em outras palavras, a proporção de dados quentes para mornos é de cerca de 20:80.
 2. Desenvolva o monitoramento do tempo ocioso do objeto no cluster.
 3. Grandes implementações de mais de 500 Gb de dados são boas candidatas.
- [Obrigatório] Entenda que as instâncias com hierarquização de dados não são opcionais para determinadas workloads.
 1. Há um pequeno custo de performance para acessar objetos usados com menos frequência, pois eles são trocados para um SSD local. Se sua aplicação for sensível ao tempo de resposta, teste o impacto na workload.
 2. Não é adequado para caches que armazenam principalmente objetos grandes com mais de 128 MiB.

[Recursos]:

- [Limitações](#)
- [Ideal] Os tipos de instância reservada oferecem suporte à hierarquização de dados. Isso garante o menor custo em termos de quantidade de armazenamento de dados por instância.
 1. Talvez seja necessário operar seus clusters do ElastiCache usando instâncias sem hierarquização de dados até ter uma melhor compreensão de seus requisitos.
 2. Analise seu padrão de uso de dados de clusters do ElastiCache.
 3. Crie um trabalho automatizado que colete periodicamente o tempo ocioso do objeto.
 4. Se você perceber que uma grande porcentagem (cerca de 80%) dos objetos fica ociosa por um período considerado apropriado para sua workload, documente as descobertas e sugira a migração do cluster para instâncias compatíveis com a hierarquização de dados.
 5. Avalie regularmente os novos tipos de nó de cache disponíveis e avalie se faz sentido, do ponto de vista das métricas de operação e custo, migrar sua frota de instâncias para novos tipos de nó de cache.

- [OBJECT IDLETIME](#)
- [Preço do Amazon ElastiCache](#)

Etapas comuns de solução de problemas e melhores práticas com ElastiCache

Os tópicos a seguir fornecem dicas de solução de problemas para erros e problemas que você pode encontrar ao usar ElastiCache. Se encontrar um problema que não esteja listado aqui, você poderá usar o botão de feedback desta página para relatá-lo.

Para obter mais orientações sobre solução de problemas e respostas a perguntas comuns de suporte, acesse a [Central de Conhecimento da AWS](#)

Tópicos

- [Problemas de conexão](#)
- [Erros do cliente Valkey ou Redis OSS](#)
- [Solução de problemas de alta latência no Serverless ElastiCache](#)
- [Solução de problemas de limitação no Serverless ElastiCache](#)
- [Problemas de conexão persistentes](#)
- [Related Topics](#)

Problemas de conexão

Se você não conseguir se conectar ao ElastiCache cache, considere uma das seguintes opções:

1. Usando o TLS: se você estiver com uma conexão interrompida ao tentar se conectar ao seu ElastiCache endpoint, talvez não esteja usando o TLS no seu cliente. Se você estiver usando o ElastiCache Serverless, a criptografia em trânsito estará sempre ativada. Certifique-se de que seu cliente esteja usando o TLS para se conectar ao cache. [Saiba mais sobre como se conectar a um cache habilitado para TLS](#).
2. VPC: ElastiCache os caches só podem ser acessados de dentro de uma VPC. Certifique-se de que a EC2 instância a partir da qual você está acessando o cache e o ElastiCache cache sejam criados na mesma VPC. Como alternativa, você deve ativar o [emparelhamento de VPC](#) entre a VPC em que sua instância EC2 reside e a VPC em que você está criando seu cache.

3. Grupos de segurança: ElastiCache usa grupos de segurança para controlar o acesso ao seu cache. Considere o seguinte:
 - a. Certifique-se de que o grupo de segurança usado pelo seu ElastiCache cache permita acesso de entrada a ele a partir da sua EC2 instância. Veja [aqui](#) para saber como configurar corretamente as regras de entrada em seu grupo de segurança.
 - b. Certifique-se de que o grupo de segurança usado pelo ElastiCache cache permita acesso às portas do cache (6379 e 6380 para clusters sem servidor e 6379 por padrão para clusters baseados em nós). ElastiCache usa essas portas para aceitar comandos Valkey ou Redis OSS. Saiba mais sobre como configurar o acesso à porta [aqui](#).

Se a conexão continuar sendo difícil, consulte [Problemas de conexão persistentes](#) para outras etapas.

Erros do cliente Valkey ou Redis OSS

ElastiCache O Serverless só pode ser acessado usando clientes que oferecem suporte ao protocolo de modo de cluster Valkey ou Redis OSS. Clusters baseados em nós podem ser acessados de clientes em qualquer um dos modos, dependendo da configuração do cluster.

Se você estiver enfrentando erros no seu cliente, considere o seguinte:

1. Modo de cluster: se você estiver enfrentando erros de CROSSLOT ou erros com o comando [SELECT](#), talvez esteja tentando acessar um cache habilitado para o modo de cluster com um cliente Valkey ou Redis OSS que não suporta o protocolo Cluster. ElastiCache O Serverless só oferece suporte a clientes que oferecem suporte ao protocolo de cluster Valkey ou Redis OSS. Se você quiser usar o Valkey ou Redis OSS no “Modo de cluster desabilitado” (CMD), será necessário criar um cluster baseado em nós.
2. Erros de CROSSLOT: Se você estiver enfrentando o erro `ERR CROSSLOT Keys in request don't hash to the same slot`, talvez esteja tentando acessar chaves que não pertencem ao mesmo slot em um cache do modo de cluster. Como lembrete, o ElastiCache Serverless sempre opera no Modo Cluster. Operações com várias chaves, transações ou scripts Lua envolvendo várias chaves são permitidos somente se todas as chaves envolvidas estiverem no mesmo slot de hash.

Para obter mais práticas recomendadas sobre a configuração de clientes Valkey ou Redis OSS, consulte esta [postagem no blog](#).

Solução de problemas de alta latência no Serverless ElastiCache

Se sua carga de trabalho parecer estar com alta latência, você pode analisar as `SuccessfulWriteRequestLatency` métricas CloudWatch `SuccessfulReadRequestLatency` e para verificar se a latência está relacionada ao Serverless. ElastiCache Essas métricas medem a latência interna ao ElastiCache Serverless - a latência do lado do cliente e os tempos de viagem da rede entre seu cliente e o endpoint ElastiCache Serverless não estão incluídos.

Solução de problemas de latência do lado do cliente

Se você notar uma latência elevada no lado do cliente, mas nenhum aumento correspondente nas `SuccessfulReadRequestLatency` `SuccessfulWriteRequestLatency` métricas que medem a latência do lado do servidor, considere o seguinte:

- Verifique se o grupo de segurança permite acesso às portas 6379 e 6380: o ElastiCache Serverless usa a porta 6379 para o endpoint primário e a porta 6380 para o endpoint do leitor. Alguns clientes estabelecem conectividade com as duas portas para cada nova conexão, mesmo que seu aplicativo não esteja usando o recurso Ler da réplica. Se o grupo de segurança não permitir acesso de entrada às duas portas, o estabelecimento da conexão poderá levar mais tempo. Saiba mais sobre como configurar o acesso à porta [aqui](#).

Solução de problemas de latência do lado do servidor

Alguma variabilidade e picos ocasionais não devem ser motivo de preocupação. No entanto, se a `Average` estatística mostrar um aumento acentuado e persistir, você deve verificar o `Personal Health Dashboard` e o `Health Dashboard` para obter mais informações. Se necessário, considere abrir um caso de suporte com Suporte.

Considere as seguintes práticas recomendadas e estratégias para reduzir a latência:

- Habilitar leitura da réplica: se seu aplicativo permitir essa opção, recomendamos habilitar o recurso “Ler da réplica” em seu cliente Valkey ou Redis OSS para escalar as leituras e obter menor latência. Quando ativado, o ElastiCache Serverless tenta rotear suas solicitações de leitura para nós de cache de réplica que estão na mesma zona de disponibilidade (AZ) do seu cliente, evitando assim a latência de rede entre AZ. Observe que, ao ativar o recurso Ler da réplica em seu cliente, seu aplicativo aceitará uma eventual consistência nos dados. Seu aplicativo talvez receba dados mais antigos por algum tempo se você tentar ler depois de gravar em uma chave.
- Certifique-se de que seu aplicativo seja implantado da AZs mesma forma que seu cache: você pode observar uma maior latência do lado do cliente se seu aplicativo não for implantado da AZs

mesma forma que seu cache. Ao criar um cache sem servidor, você pode fornecer as sub-redes de onde seu aplicativo acessará o cache, e o Serverless ElastiCache cria VPC Endpoints nessas sub-redes. Certifique-se de que seu aplicativo seja implantado no mesmo AZs. Caso contrário, seu aplicativo poderá sofrer um salto entre AZ ao acessar o cache, resultando em maior latência do lado do cliente.

- **Conexões de reutilização:** as solicitações ElastiCache sem servidor são feitas por meio de uma conexão TCP habilitada para TLS usando o protocolo RESP. Iniciar a conexão (incluindo a autenticação da conexão, se configurada) leva tempo, então a latência da primeira solicitação é maior que o normal. Solicitações em uma conexão já inicializada oferecem ElastiCache baixa latência consistente. Por esse motivo, você deve considerar o uso do pool de conexões ou a reutilização das conexões OSS Valkey ou Redis existentes.
- **Velocidade de escalonamento:** o ElastiCache Serverless é escalado automaticamente à medida que sua taxa de solicitações aumenta. Um grande aumento repentino na taxa de solicitações, mais rápido do que a velocidade na qual o ElastiCache Serverless é escalado, pode resultar em latência elevada por algum tempo. ElastiCache Normalmente, o Serverless pode aumentar rapidamente a taxa de solicitações suportadas, levando de 10 a 12 minutos para dobrar a taxa de solicitações.
- **Inspecione comandos de longa execução:** alguns comandos do Valkey ou do Redis OSS, incluindo scripts Lua ou comandos em grandes estruturas de dados, podem ser executados por muito tempo. Para identificar esses comandos, ElastiCache publica métricas de nível de comando. Com o [ElastiCache Serverless](#), você pode usar as BasedECPUs métricas.
- **Solicitações limitadas:** quando as solicitações são limitadas no ElastiCache Serverless, você pode experimentar um aumento na latência do lado do cliente em seu aplicativo. [Quando as solicitações são limitadas no ElastiCache Serverless, você deve ver um aumento na métrica Serverless. ThrottledRequests ElastiCache](#) Consulte a seção abaixo para solucionar problemas em solicitações com controle de utilização.
- **Distribuição uniforme de chaves e solicitações:** no ElastiCache Valkey e no Redis OSS, uma distribuição desigual de chaves ou solicitações por slot pode resultar em um hot slot, o que pode resultar em latência elevada. ElastiCache O Serverless suporta até 30.000 ECPUs/second (90.000 ECPUs/second ao usar Read from Replica) em um único slot, em uma carga de trabalho que executa comandos simples. SET/GET Recomendamos avaliar a distribuição da chave e da solicitação em todos os slots e garantir uma distribuição uniforme se a taxa de solicitação exceder esse limite.

Solução de problemas de limitação no Serverless ElastiCache

Em arquiteturas orientadas a serviços e sistemas distribuídos, a limitação da taxa na qual as chamadas de API são processadas por vários componentes do serviço é chamada de controle de utilização. Isso suaviza os picos, controla as incompatibilidades na produtividade dos componentes e permite recuperações mais previsíveis quando há um evento operacional inesperado. ElastiCache O Serverless foi projetado para esses tipos de arquiteturas, e a maioria dos clientes Valkey ou Redis OSS tem novas tentativas incorporadas para solicitações limitadas. Algum grau de controle de utilização não é necessariamente um problema para a aplicação, mas o controle de utilização persistente de uma parte sensível à latência do fluxo de trabalho de dados pode afetar negativamente a experiência do usuário e reduzir a eficiência geral do sistema.

[Quando as solicitações são limitadas no ElastiCache Serverless, você deve ver um aumento na métrica Serverless. ThrottledRequests ElastiCache](#) Se você está percebendo um grande número de solicitações com controle de utilização, considere o seguinte:

- **Velocidade de escalabilidade:** o ElastiCache Serverless é escalado automaticamente à medida que você ingere mais dados ou aumenta sua taxa de solicitações. Se seu aplicativo for dimensionado mais rápido do que a velocidade com que o Serverless é escalado, suas solicitações podem ser limitadas, enquanto o ElastiCache Serverless é escalado para acomodar sua carga de trabalho. ElastiCache Normalmente, a tecnologia sem servidor pode aumentar o tamanho do armazenamento rapidamente, levando de 10 a 12 minutos para dobrar o tamanho do armazenamento em seu cache.
- **Distribuição uniforme de chaves e solicitações:** no ElastiCache Valkey e no Redis OSS, uma distribuição desigual de chaves ou solicitações por slot pode resultar em um hot slot. Um hot slot pode resultar na limitação de solicitações, se a taxa de solicitações para um único slot exceder 30.000 ECPUs/second e estiver em uma carga de trabalho que executa comandos simples. SET/GET Da mesma forma, com ElastiCache o Memcached, uma tecla de atalho pode resultar na limitação de solicitações se a taxa de solicitação exceder 30.000 por segundo. ECPUs
- **Ler da réplica:** se sua aplicação permitir, considere usar o recurso “Ler da réplica”. A maioria dos clientes Valkey ou Redis OSS pode ser configurada para “escalar leituras” para direcionar as leituras para os nós de réplica. Esse recurso permite que você escale o tráfego de leitura. Além disso, o ElastiCache Serverless encaminha automaticamente a leitura das solicitações de réplica para os nós na mesma zona de disponibilidade do seu aplicativo, resultando em menor latência. Quando a opção Ler da réplica está ativada, você pode alcançar até 90.000 ECPUs/second em um único slot, para cargas de trabalho com comandos simples. SET/GET

Problemas de conexão persistentes

Os seguintes itens devem ser verificados ao solucionar problemas persistentes de conectividade com o ElastiCache:

Tópicos

- [Grupos de segurança](#)
- [Network ACLs](#)
- [Tabelas de rotas](#)
- [Resolução do DNS](#)
- [Identificação de problemas com o diagnóstico do lado do servidor](#)
- [Validação da conectividade de rede](#)
- [Limites relacionados à rede](#)
- [Uso da CPU](#)
- [Conexões sendo encerradas do lado do servidor](#)
- [Solução de problemas no lado do cliente para instâncias do Amazon EC2](#)
- [Dissecação do tempo necessário para concluir uma única solicitação](#)

Grupos de segurança

Os grupos de segurança são firewalls virtuais que protegem o cliente do ElastiCache (instância do EC2, função AWS Lambda, contêiner do Amazon ECS etc.) e cache do ElastiCache. Os grupos de segurança têm estado, o que significa que depois que o tráfego de entrada ou saída é permitido, as respostas para esse tráfego serão automaticamente autorizadas no contexto desse grupo de segurança específico.

O recurso stateful (com estado) requer que o grupo de segurança mantenha o controle de todas as conexões autorizadas e há um limite para conexões controladas. Se o limite for atingido, as novas conexões apresentarão falha. Consulte a seção de solução de problemas para obter ajuda sobre como identificar se os limites foram atingidos do lado do cliente ou do ElastiCache.

Você pode ter um único grupo de segurança atribuído ao mesmo tempo ao cliente e ao cluster do ElastiCache, ou grupos de segurança individuais para cada um.

Em ambos os casos, você precisa permitir o tráfego de saída TCP na porta do ElastiCache da origem e o tráfego de entrada na mesma porta para o ElastiCache. A porta padrão é 11211 para

Memcached e 6379 para Valkey ou Redis OSS. Por padrão, os grupos de segurança permitem todo o tráfego de saída. Nesse caso, somente a regra de entrada no grupo de segurança de destino é necessária.

Para obter mais informações, consulte [Padrões de acesso para acessar um cluster do ElastiCache em uma Amazon VPC](#).

Network ACLs

As listas de controle de acesso à rede (ACLs) são regras sem estado. O tráfego deve ser permitido em ambas as direções (entrada e saída) para ter êxito. As ACLs de rede são atribuídas a sub-redes, não a recursos específicos. É possível ter a mesma ACL atribuída ao ElastiCache e ao recurso do cliente, especialmente se eles estiverem na mesma sub-rede.

Por padrão, as ACLs de rede permitem todo o tráfego. No entanto, é possível personalizá-las para negar ou permitir tráfego. Além disso, a avaliação das regras da ACL é sequencial, o que significa que a regra com o menor número correspondente ao tráfego irá permiti-lo ou negá-lo. A configuração mínima para permitir o tráfego Valkey ou Redis OSS é:

ACL de rede do lado do cliente:

- Regras de entrada:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Type: Custom TCP Rule;
- Protocolo: TCP
- Port Range: 1024-65535
- Source: 0.0.0.0/0 (ou criar regras individuais para as sub-redes do cluster do ElastiCache)
- Allow/Deny: Allow
- Regras de saída:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Type: Custom TCP Rule;
- Protocolo: TCP
- Port Range: 6379
- Source 0.0.0.0/0 (ou as sub-redes do cluster do ElastiCache. Tenha em mente que o uso de IPs específicos pode criar problemas em caso de failover ou escalabilidade do cluster)

- Allow/Deny: Allow

ACL de rede do ElastiCache:

- Regras de entrada:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Type: Custom TCP Rule;
- Protocolo: TCP
- Port Range: 6379
- Source: 0.0.0.0/0 (ou criar regras individuais para as sub-redes do cluster do ElastiCache)
- Allow/Deny: Allow
- Regras de saída:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Type: Custom TCP Rule;
- Protocolo: TCP
- Port Range: 1024-65535
- Source 0.0.0.0/0 (ou as sub-redes do cluster do ElastiCache. Tenha em mente que o uso de IPs específicos pode criar problemas em caso de failover ou escalabilidade do cluster)
- Allow/Deny: Allow

Para obter mais informações, consulte [ACLs de rede](#).

Tabelas de rotas

Da mesma forma que as ACLs de rede, cada sub-rede pode ter tabelas de rota diferentes. Se os clientes e o cluster do ElastiCache estiverem em sub-redes diferentes, certifique-se de que suas tabelas de rota permitam que eles cheguem uns aos outros.

Ambientes mais complexos, envolvendo várias VPCs, roteamento dinâmico ou firewalls de rede, podem se tornar difíceis para a solução de problemas. Consulte [Validação da conectividade de rede](#) para confirmar se as configurações de rede são apropriadas.

Resolução do DNS

O ElastiCache fornece os endpoints de serviço com base em nomes do DNS. Os endpoints disponíveis são os endpoints Configuration, Primary, Reader, e Node. Para obter mais informações, consulte [Encontrar endpoints de conexão](#).

Em caso de failover ou modificação de cluster, o endereço associado ao nome do endpoint pode mudar e será atualizado automaticamente.

Configurações de DNS personalizadas (ou seja, não usando o serviço de DNS da VPC) podem não estar cientes dos nomes de DNS fornecidos pelo ElastiCache. Certifique-se de que seu sistema possa resolver com êxito os endpoints do ElastiCache usando ferramentas do sistema como o `dig` (como mostrado a seguir) ou o `nslookup`.

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Você também pode forçar a resolução de nomes por meio do serviço de DNS da 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
```

Identificação de problemas com o diagnóstico do lado do servidor

As métricas do CloudWatch e as informações de tempo de execução do mecanismo ElastiCache são fontes ou informações comuns para identificar potenciais fontes de problemas de conexão. Uma boa análise geralmente começa com os seguintes itens:

- **Uso da CPU:** Valkey e Redis OSS são aplicativos multithread. No entanto, a execução de cada comando ocorre em um único thread (principal). Por esse motivo, o ElastiCache fornece as métricas `CPUUtilization` e `EngineCPUUtilization`. `EngineCPUUtilization` fornece a utilização da CPU dedicada ao processo do Valkey ou Redis OSS, e `CPUUtilization`, o uso em todas as vCPUs. Os nós com mais de uma vCPU geralmente têm valores diferentes para `CPUUtilization` e `EngineCPUUtilization`, sendo o segundo comumente maior. Um `EngineCPUUtilization` alto pode ser causado por um número elevado de solicitações ou operações complexas que levem uma quantidade significativa de tempo da CPU para ser concluída. Você pode identificar ambos com o seguinte:

- Número elevado de solicitações: verifique se há aumentos em outras métricas que correspondam ao padrão `EngineCPUUtilization`. As métricas úteis são:
 - `CacheHits` e `CacheMisses`: o número de solicitações ou solicitações bem-sucedidas que não encontraram um item válido no cache. Se a proporção de erros em comparação com acertos for alta, a aplicação está desperdiçando tempo e recursos com solicitações infrutíferas.
 - `SetTypeCmds` e `GetTypeCmds`: Essas métricas correlacionadas com `EngineCPUUtilization` podem ajudar a entender se a carga é significativamente maior para solicitações de gravação, medida por `SetTypeCmds`, ou leituras, medido por `GetTypeCmds`. Se a carga for predominantemente de leituras, o uso de várias réplicas de leitura pode equilibrar as solicitações em vários nós e poupar o primário para gravações. Em clusters com modo cluster desabilitado, o uso de réplicas de leitura pode ser feito criando uma configuração de conexão adicional na aplicação usando o endpoint leitor do ElastiCache. Para obter mais informações, consulte [Encontrar endpoints de conexão](#). As operações de leitura devem ser enviadas para essa conexão adicional. As operações de gravação serão feitas através do endpoint primário regular. No modo cluster habilitado, é aconselhável usar uma biblioteca com suporte a réplicas de leitura nativamente. Com os sinalizadores certos, a biblioteca será capaz de descobrir automaticamente a topologia do cluster, os nós de réplica, habilitar as operações de leitura através do comando `READONLY` do Valkey Redis OSS e enviar as solicitações de leitura para as réplicas.
- Número elevado de conexões:
 - `CurrConnections` e `NewConnections`: `CurrConnection` é o número de conexões estabelecidas no momento da coleta de pontos de dados, enquanto `NewConnections` mostra quantas conexões foram criadas no período.

Criar e manipular conexões implica em sobrecarga significativa da CPU. Além disso, o handshake de três vias de TCP necessário para criar novas conexões afetará negativamente os tempos de resposta gerais.

Um nó do ElastiCache com milhares de `NewConnections` por minuto indica que uma conexão é criada e usada por apenas alguns comandos, o que não é ideal. Manter as conexões estabelecidas e reutilizá-las para novas operações é uma prática recomendada. Isso é possível quando a aplicação cliente oferece suporte e implementa corretamente o grupo de conexões ou as conexões persistentes. Com o grupo de conexões, o número de `currConnections` não tem grandes variações, e o `NewConnections` deve ser o mais baixo possível. Valkey e Redis OSS oferecem desempenho ideal com um pequeno número de

`currConnections`. Manter `currConnection` na ordem de dezenas ou centenas minimiza o uso de recursos para suportar conexões individuais, como buffers de cliente e ciclos de CPU para servir a conexão.

- **Throughput da rede**
 - Determine a largura de banda: os nós do ElastiCache têm largura de banda proporcional ao tamanho do nó. Como as aplicações têm características diferentes, os resultados podem variar de acordo com a workload. Como exemplos, aplicações com alta taxa de pequenas solicitações tendem a afetar mais o uso da CPU do que o throughput da rede, enquanto chaves maiores causarão maior utilização da rede. Por esse motivo, é aconselhável testar os nós com a workload real para uma melhor compreensão dos limites.

A simulação da carga da aplicação forneceria resultados mais precisos. No entanto, as ferramentas de benchmark podem dar uma boa ideia dos limites.

- Para casos em que as solicitações são predominantemente leituras, o uso de réplicas para operações de leitura aliviará a carga no nó primário. Se o caso de uso for predominantemente gravações, o uso de muitas réplicas amplificará o uso da rede. Para cada byte gravado no nó primário, N bytes serão enviados para as réplicas, sendo N o número de réplicas. A prática recomendada para workloads intensas de gravação é usar o ElastiCache para Redis OSS com modo cluster habilitado para que as gravações possam ser balanceadas em vários fragmentos ou haver aumento de escala vertical para um tipo de nó com mais recursos de rede.
- As métricas `NetworkBytesIn` e `NetworkBytesOut` do Cloudwatch fornecem a quantidade de dados que entram ou saem do nó, respectivamente. `ReplicationBytes` é o tráfego dedicado à replicação de dados.

Para obter mais informações, consulte [Limites relacionados à rede](#).

- **Comandos complexos:** os comandos do Redis OSS são servidos em um único thread, o que significa que as solicitações são atendidas sequencialmente. Um único comando lento pode afetar outras solicitações e conexões, culminando em tempos limite. O uso de comandos que atuem sobre vários valores, chaves ou tipos de dados deve ser feito com cuidado. As conexões podem ser bloqueadas ou terminadas dependendo do número de parâmetros ou do tamanho de seus valores de entrada ou saída.

Um exemplo notório é o comando `KEYS`. Ele varre todo o keyspace procurando por um determinado padrão e bloqueia a execução de outros comandos durante sua execução. O Redis OSS usa a notação "Big O" para descrever sua complexidade de comandos.

O comando `Keys` tem complexidade de tempo $O(N)$, sendo N o número de chaves no banco de dados. Portanto, quanto maior o número de chaves, mais lento será o comando. `KEYS` pode causar problemas de maneiras diferentes: se nenhum padrão de pesquisa for usado, o comando retornará todos os nomes de chaves disponíveis. Em bancos de dados com milhares ou milhões de itens, uma enorme saída será criada e inundará os buffers de rede.

Se um padrão de pesquisa for usado, somente as chaves correspondentes ao padrão retornarão ao cliente. No entanto, o mecanismo ainda varrerá todo o `keyspace` procurando por ele, e o tempo para concluir o comando será o mesmo.

Uma alternativa a `KEYS` é o comando `SCAN`. Ele itera sobre o `keyspace` e limita as iterações em um número específico de itens, evitando bloqueios prolongados no mecanismo.

A varredura tem o parâmetro `COUNT`, usado para definir o tamanho dos blocos de iteração. O valor padrão é 10 (10 itens por iteração).

Dependendo do número de itens no banco de dados, blocos com pequenos valores de `COUNT` irão exigir mais iterações para concluir uma verificação completa, e valores maiores manterão o mecanismo ocupado por mais tempo em cada iteração. Enquanto pequenos valores de contagem farão `SCAN` mais lento em grandes bancos de dados, valores maiores podem causar os mesmos problemas mencionados para `KEYS`.

Como exemplo, a execução do comando `SCAN` com valor de contagem de 10 vai requer 100.000 repetições em um banco de dados com 1 milhão de chaves. Se o tempo médio de ida e volta da rede for de 0,5 milissegundos, aproximadamente 50.000 milissegundos (50 segundos) serão gastos transferindo solicitações.

Por outro lado, se o valor da contagem fosse 100.000, uma única iteração seria necessária e apenas 0,5 ms seriam gastos transferindo-a. No entanto, o mecanismo seria totalmente bloqueado para outras operações até que o comando terminasse de varrer todo o `keyspace`.

Além de `KEYS`, vários outros comandos são potencialmente prejudiciais se não forem usados corretamente. Para ver uma lista de todos os comandos e suas respectivas complexidades de tempo, acesse [Comandos Valkey e Redis OSS](#).

Exemplos de possíveis problemas:

- **Scripts Lua:** o Valkey e Redis OSS fornecem um interpretador Lua incorporado, permitindo a execução de scripts no lado do servidor. Os scripts Lua no Valkey e no Redis OSS são

executados no nível do mecanismo e são atômicos por definição, o que significa que nenhum outro comando ou script terá execução permitida enquanto um script estiver em execução. Os scripts Lua fornecem a possibilidade de executar vários comandos, algoritmos de tomada de decisão, análise de dados e outros diretamente no mecanismo. Embora a atomicidade dos scripts e a chance de descarregar a aplicação sejam tentadoras, os scripts devem ser usados com cuidado e para pequenas operações. No ElastiCache, o tempo de execução dos scripts Lua é limitado a 5 segundos. Scripts que não forem gravados no keyspace serão encerrados automaticamente após o período de 5 segundos. Para evitar a corrupção de dados e inconsistências, o nó fará failover se a execução do script não tiver sido concluída em 5 segundos e tiver qualquer gravação durante sua execução. [Transações](#) são a alternativa para garantir a consistência de múltiplas modificações de chaves relacionadas no Redis OSS. Uma transação permite a execução de um bloco de comandos, observando as chaves existentes para modificações. Se qualquer uma das chaves observadas mudar antes da conclusão da transação, todas as modificações serão descartadas.

- Exclusão em massa de itens: o comando DEL aceita vários parâmetros, que são os nomes das chaves a serem excluídas. As operações de exclusão são síncronas e gastarão um tempo significativo da CPU se a lista de parâmetros for grande ou contiver uma grande lista, conjunto, conjunto ordenado ou hash (estruturas de dados contendo vários subitens). Em outras palavras, até mesmo a exclusão de uma única chave pode levar um tempo significativo se tiver muitos elementos. A alternativa a DEL é UNLINK, que é um comando assíncrono disponível desde o Redis OSS 4. UNLINK deve ter preferência sobre DEL sempre que possível. Começando no ElastiCache para Redis OSS 6x, o parâmetro `lazyfree-lazy-user-del` faz com que o parâmetro DEL se comporte como UNLINK quando habilitado. Para mais informações, consulte [Alterações de parâmetro do Redis OSS 6.0](#).
- Comandos que atuam sobre várias chaves: DEL foi mencionado anteriormente como um comando que aceita vários argumentos e seu tempo de execução será diretamente proporcional a isso. No entanto, o Redis OSS fornece muitos outros comandos que funcionam de forma semelhante. Como exemplos, MSET e MGET permitem a inserção ou recuperação de várias chaves de string ao mesmo tempo. Seu uso pode ser benéfico para reduzir a latência de rede inerente a vários comandos individuais SET ou GET. No entanto, uma extensa lista de parâmetros afetará o uso da CPU.

Embora a utilização da CPU por si só não seja a causa de problemas de conectividade, gastar muito tempo para processar um único ou poucos comandos através de várias chaves pode causar falha de outras solicitações e aumentar a utilização geral da CPU.

O número de chaves e seu tamanho afetarão a complexidade do comando e, consequentemente, o tempo de conclusão.

Outros exemplos de comandos que podem atuar sobre várias chaves: HMGET, HMSET, MSETNX, PFCOUNT, PFMERGE, SDIFF, SDIFFSTORE, SINTER, SINTERSTORE, SUNION, SUNIONSTORE, TOUCH, ZDIFF, ZDIFFSTORE, ZINTER ou ZINTERSTORE.

- Comandos atuando sobre vários tipos de dados: o Redis OSS também fornece comandos que atuam em uma ou várias chaves, independentemente do tipo de dados. O ElastiCache para Redis OSS fornece a métrica KeyBasedCmds para monitorar esses comandos. Essa métrica soma a execução dos seguintes comandos no período selecionado:
 - Complexidade de $O(N)$:
 - KEYS
 - $O(1)$
 - EXISTS
 - OBJECT
 - PTTL
 - RANDOMKEY
 - TTL
 - TYPE
 - EXPIRE
 - EXPIREAT
 - MOVE
 - PERSIST
 - PEXPIRE
 - PEXPIREAT
 - UNLINK ($O(N)$) para recuperar memória. No entanto, a tarefa de recuperação de memória ocorre em um thread separado e não bloqueia o mecanismo
 - Diferentes tempos de complexidade dependendo do tipo de dados:
 - DEL
 - DUMP

- RENAME é considerado um comando com complexidade $O(1)$, mas executa DEL internamente. O tempo de execução varia de acordo com o tamanho da chave renomeada.
- RENAMENX
- RESTORE
- SORT
- Hashes grandes: hash é um tipo de dado que permite uma única chave com vários subitens de valor-chave. Cada hash pode armazenar 4.294.967.295 itens e operações em hashes grandes podem se tornar caras. Da mesma forma que KEYS, hashes têm o comando HKEYS com complexidade de tempo $O(N)$, N sendo o número de itens no hash. HSCAN deve ter preferência sobre HKEYS para evitar comandos de longa execução. HDEL, HGETALL, HMGET, HMSET e HVALS são comandos que devem ser usados com cautela em hashes grandes.
- Outras estruturas grandes de dados: além de hashes, outras estruturas de dados podem ser intensas em CPU. Conjuntos, listas, conjuntos classificados e HyperLogLogs também podem levar um tempo significativo para serem manipulados, dependendo do tamanho e dos comandos usados. Para obter mais informações sobre esses comandos, consulte [Comandos Valkey e Redis OSS](#).

Validação da conectividade de rede

Depois de analisar as configurações de rede relacionadas à resolução de DNS, grupos de segurança, ACLs de rede e tabelas de rota, a conectividade pode ser validada com o VPC Reachability Analyzer e as ferramentas do sistema.

O Reachability Analyzer testará a conectividade de rede e confirmará se todos os requisitos e permissões estão satisfeitos. Para os testes abaixo, você precisará do ID da ENI (Identificação de interface de rede elástica) de um dos nós do ElastiCache disponíveis em sua VPC. É possível fazer isso da seguinte maneira:

1. Acesse <https://console.aws.amazon.com/ec2/v2/home?#NIC:>
2. Filtre a lista de interfaces pelo nome do cluster do ElastiCache ou pelo endereço IP obtido das validações de DNS anteriormente.
3. Anote ou salve o ID da ENI. Se várias interfaces forem exibidas, revise a descrição para confirmar que elas pertencem ao cluster correto do ElastiCache e escolha uma delas.

4. Prossiga para a próxima etapa.
5. Crie um caminho de análise em <https://console.aws.amazon.com/vpc/home?#ReachabilityAnalyzer> e escolha as seguintes opções:
 - Source Type (Tipo de origem): escolha Instance (Instância) se seu cliente ElastiCache for executado em uma instância do Amazon EC2, ou Network Interface (Interface de rede) se ele usar outro serviço, como o Amazon ECS AWS Fargate com rede awsvpc, AWS Lambda, etc.), e o respectivo ID de recurso (instância EC2 ou ID da ENI);
 - Destination Type (Tipo de destino): escolha Network Interface (Interface de rede) e selecione a ElastiCache ENI (ENI do ElastiCache) na lista.
 - Destination port (Porta de destino): especifique 6379 para o ElastiCache para Redis OSS ou 11211 para o ElastiCache para Memcached. Essas são as portas definidas com a configuração padrão e este exemplo pressupõe que elas não foram alteradas.
 - Protocolo: TCP

Crie o caminho de análise e aguarde alguns instantes para o resultado. Se o status estiver inacessível, abra os detalhes da análise e revise Analysis Explorer (Explorador da análise) para obter detalhes sobre onde as solicitações foram bloqueadas.

Se os testes de acessibilidade forem aprovados, avance para a verificação no nível do sistema operacional.

Para validar a conectividade de TCP na porta de serviço do ElastiCache: no Amazon Linux, Nping está disponível no pacote nmap e pode testar a conectividade de TCP na porta do ElastiCache, além de fornecer o tempo de ida e volta da rede para estabelecer a conexão. Use isso para validar a conectividade de rede e a latência atual para o cluster do ElastiCache, conforme mostrado a seguir:

```
$ 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
```

Por padrão, `nping` envia 5 testes com um atraso de 1 segundo entre eles. Você pode usar a opção `-c` para aumentar o número de testes e `--delay` para alterar o tempo de envio de um novo teste.

Se os testes com `nping` falharem e os testes do VPC Reachability Analyzer forem aprovados, peça ao administrador do sistema para revisar possíveis regras de firewall baseadas em host, regras de roteamento assimétricas ou qualquer outra restrição possível no nível do sistema operacional.

No console do ElastiCache, verifique se Encryption in-transit (Criptografia em trânsito) está habilitada nos detalhes do cluster do ElastiCache. Se a criptografia em trânsito estiver habilitada, confirme se a sessão de TLS pode ser estabelecida com o seguinte comando:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Uma saída extensa é esperada se a conexão e a negociação de TLS forem bem-sucedidas.

Verifique o código de retorno disponível na última linha, o valor deve ser de 0 (ok). Se o `openssl` retornar algo diferente, verifique o motivo do erro em <https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS>.

Se todos os testes de infraestrutura e sistema operacional forem aprovados, mas sua aplicação ainda não conseguir se conectar ao ElastiCache, verifique se as configurações da aplicação estão em conformidade com as configurações do ElastiCache. Erros comuns são:

- Sua aplicação não suporta o modo de cluster do ElastiCache e o ElastiCache tem modo cluster habilitado;
- Sua aplicação não oferece suporte a TLS/SSL e o ElastiCache possui criptografia em trânsito habilitada;
- A aplicação suporta TLS/SSL, mas não possui os sinalizadores de configuração corretos ou autoridades de certificação confiáveis;

Limites relacionados à rede

- Número máximo de conexões: há limites rígidos para conexões simultâneas. Cada nó do ElastiCache permite até 65.000 conexões simultâneas em todos os clientes. Este limite pode ser monitorado através das métricas `CurrConnections` métricas no CloudWatch. No entanto, os clientes também têm seus limites para conexões de saída. No Linux, verifique o intervalo de portas efêmeras permitido com o comando:

```
# sysctl net.ipv4.ip_local_port_range
```

```
net.ipv4.ip_local_port_range = 32768 60999
```

No exemplo anterior, 28231 conexões serão permitidas da mesma origem, para o mesmo IP de destino (nó ElastiCache) e porta. O comando a seguir mostra quantas conexões existem para um nó do ElastiCache específico (IP 1.2.3.4):

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE  
'^State' | wc -l
```

Se o número for muito alto, seu sistema pode ficar sobrecarregado tentando processar as solicitações de conexão. É aconselhável considerar a implementação de técnicas como grupo de conexões ou conexões persistentes para lidar melhor com as conexões. Sempre que possível, configure o grupo de conexões para limitar o número máximo de conexões a algumas centenas. Além disso, a lógica de recuo para lidar com tempo limite ou outras exceções de conexão seria aconselhável para evitar rotatividade de conexão em caso de problemas.

- Limites de tráfego de rede: verifique as seguintes [métricas do CloudWatch para o Redis OSS](#) para identificar possíveis limites de rede que estejam sendo atingidos no nó do ElastiCache:
 - `NetworkBandwidthInAllowanceExceeded/NetworkBandwidthOutAllowanceExceeded`: pacotes de rede modelados porque o throughput excedeu o limite de largura de banda agregada.

É importante observar que cada byte gravado no nó primário será replicado para N réplicas, sendo N o número de réplicas. Clusters com tipos de nó pequenos, várias réplicas e solicitações de gravação intensiva podem não conseguir lidar com o backlog de replicação. Nesses casos, é uma prática recomendada aumentar a escala vertical (alterar o tipo de nó), aumentar a escala horizontal (adicionar fragmentos em clusters com modo cluster habilitado), reduzir o número de réplicas ou minimizar o número de gravações.

- `NetworkConntrackAllowanceExceeded`: pacotes modelados porque o número máximo de conexões monitoradas em todos os grupos de segurança atribuídos ao nó foi excedido. Novas conexões provavelmente falharão durante esse período.
- `NetworkPackets PerSecondAllowanceExceeded`: número máximo de pacotes por segundo excedido. Workloads baseadas em uma alta taxa de solicitações muito pequenas podem atingir esse limite antes da largura de banda máxima.

As métricas acima são a maneira ideal de confirmar que os nós atingem seus limites de rede. No entanto, os limites também são identificáveis por platôs em métricas de rede.

Se os platôs forem observados por períodos prolongados, eles provavelmente serão seguidos por atraso de replicação, aumento de bytes usados para cache, queda de memória livre, alto swap e uso da CPU. As instâncias do Amazon EC2 também têm limites de rede que podem ser rastreados por meio de [Métricas do driver ENA](#). Instâncias do Linux com suporte de rede aprimorado e drivers ENA 2.2.10 ou mais recentes podem revisar os contadores de limite com o comando:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

Uso da CPU

A métrica de uso da CPU é o ponto de partida da investigação, e os seguintes itens podem ajudar a reduzir possíveis problemas no lado do ElastiCache:

- **SlowLogs do Redis OSS:** a configuração padrão do ElastiCache retém os últimos 128 comandos que levaram mais de 10 milissegundos para serem concluídos. O histórico de comandos lentos é mantido durante o tempo de execução do mecanismo e será perdido em caso de falha ou reinicialização. Se a lista atingir 128 entradas, eventos antigos serão removidos para abrir espaço para novas entradas. O tamanho da lista de eventos lentos e o tempo de execução considerado lento podem ser modificados através dos parâmetros `slowlog-max-len` e `slowlog-log-slower-than` em um [grupo de parâmetros personalizado](#). A lista de logs lentos pode ser recuperada executando `SLOWLOG GET 128` no mecanismo, 128 sendo os últimos 128 comandos lentos relatados. Cada entrada tem os seguintes campos:

```
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
```

O evento acima aconteceu em 26 de dezembro, às 19:26:07 UTC, levou 4,8 segundos (4,823 ms) para ser concluído e foi causado pelo comando KEYS solicitado pelo cliente 1.2.3.4.

No Linux, o carimbo de data/hora pode ser convertido com a data do comando:

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Com Python:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

Ou no Windows com a 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
```

Muitos comandos lentos em um curto período de tempo (mesmo minuto ou menos) são motivo de preocupação. Revise a natureza dos comandos e como eles podem ser otimizados (consulte exemplos anteriores). Se comandos com complexidade de tempo $O(1)$ são frequentemente relatados, verifique os outros fatores para o alto uso da CPU mencionado anteriormente.

- **Métricas de latência:** o ElastiCache para Redis OSS fornece métricas do CloudWatch para monitorar a latência média de diferentes classes de comandos. O ponto de dados é calculado dividindo o número total de execuções de comandos na categoria pelo tempo total de execução no período. É importante entender que os resultados da métrica de latência são um agregado de vários comandos. Um único comando pode causar resultados inesperados, como tempos limite, sem impacto significativo nas métricas. Para tais casos, os eventos de log lento seriam uma

fonte de informação mais precisa. A lista a seguir contém as métricas de latência disponíveis e os respectivos comandos que as afetam.

- **EvalBasedCmdsLatency**: relacionado a comandos de scripts Lua, `eval`, `evalsha`;
- **GeoSpatialBasedCmdsLatency**: `geodist`, `geohash`, `geopos`, `georadius`, `georadiusbymember`, `geoadd`;
- **GetTypeCmdsLatency**: comandos de leitura, independentemente do tipo de dados;
- **HashBasedCmdsLatency**: `hexists`, `hget`, `hgetall`, `hkeys`, `hlen`, `hmget`, `hvals`, `hstrlen`, `hdel`, `hincrby`, `hincrbyfloat`, `hset`, `hsetnx`;
- **HyperLogLogBasedCmdsLatency**: `pfselftest`, `pfcount`, `pfdebug`, `pfadd`, `pfmerge`;
- **KeyBasedCmdsLatency**: comandos que podem atuar em diferentes tipos de dados: `dump`, `exists`, `keys`, `object`, `pttl`, `randomkey`, `ttd`, `type`, `del`, `expire`, `expireat`, `move`, `persist`, `pexpire`, `pexpireat`, `rename`, `renamenx`, `restoreK`, `sort`, `unlink`;
- **ListBasedCmdsLatency**: `lindex`, `llen`, `lrange`, `lpop`, `brpop`, `brpoplpush`, `linsert`, `lpop`, `lpush`, `lpushx`, `lrem`, `lset`, `ltrim`, `rpop`, `rpoplpush`, `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**: comandos de gravação, independentemente do tipo de dados;
- **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`;
- **Comandos de runtime do Redis OSS**:
 - `info commandstats`: fornece uma lista de comandos executados desde que o mecanismo foi iniciado, seu número de execuções cumulativas, tempo total de execução e tempo médio de execução por comando;

- **client list:** fornece uma lista de clientes atualmente conectados e informações relevantes como o uso de buffers, último comando executado, etc.;
- **Backup e replicação:** as versões do ElastiCache para Redis OSS anteriores à 2.8.22 usam um processo bifurcado para criar backups e processar sincronizações completas com as réplicas. Esse método pode incorrer em sobrecarga de memória significativa para casos de uso intensivo de gravação.

Começando com ElastiCache Redis OSS 2.8.22, a AWS introduziu um método de backup e replicação sem bifurcação. O novo método pode atrasar as gravações para evitar falhas. Ambos os métodos podem causar períodos de maior utilização da CPU, levar a tempos de resposta mais altos e, conseqüentemente, levar a estouro de limites de tempo do cliente durante a execução. Sempre verifique se as falhas do cliente acontecem durante a janela de backup ou a métrica `SaveInProgress` foi 1 no período. É aconselhável agendar a janela de backup para períodos de baixa utilização para minimizar a possibilidade de problemas com clientes ou falhas de backup.

Conexões sendo encerradas do lado do servidor

A configuração padrão do ElastiCache para Redis OSS mantém as conexões do cliente estabelecidas indefinidamente. No entanto, em alguns casos, o encerramento da conexão pode ser desejável. Por exemplo:

- Bugs na aplicação do cliente podem fazer com que as conexões sejam esquecidas e mantidas estabelecidas com um estado ocioso. Isso é chamado de "vazamento de conexão", e a consequência é um aumento constante no número de conexões estabelecidas observadas na métrica `CurrConnections`. Esse comportamento pode resultar em saturação no lado do cliente ou do ElastiCache. Quando uma correção imediata não é possível do lado do cliente, alguns administradores definem um valor de "tempo limite" em seu grupo de parâmetros do ElastiCache. O tempo limite é o tempo em segundos permitido para que as conexões ociosas persistam. Se o cliente não enviar qualquer solicitação no período, o mecanismo encerrará a conexão assim que a conexão atingir o valor de tempo limite. Pequenos valores de tempo limite podem resultar em desconexões desnecessárias e os clientes precisarão lidar com eles corretamente e reconectar, causando atrasos.
- A memória usada para armazenar chaves é compartilhada com buffers do cliente. Clientes lentos com grandes solicitações ou respostas podem exigir uma quantidade significativa de memória para lidar com seus buffers. As configurações padrão do ElastiCache para Redis OSS não restringem o tamanho dos buffers de saída do cliente regulares. Se o limite `maxmemory` for

atingido, o mecanismo tentará despejar itens para cumprir o uso do buffer. Em condições de memória extremamente baixa, o ElastiCache para Redis OSS pode optar por desconectar clientes que consumam grandes buffers de saída de cliente para liberar memória e reter a integridade do cluster.

É possível limitar o tamanho dos buffers do cliente com configurações personalizadas e os clientes que atingirem o limite serão desconectados. No entanto, os clientes devem ser capazes de lidar com desconexões inesperadas. Os parâmetros para lidar com o tamanho de buffers para clientes regulares são os seguintes:

- `cliente-query-buffer-limit`: tamanho máximo de uma única solicitação de entrada;
- `client-output-buffer-limit-normal-soft-limit`: limite flexível para conexões de clientes. A conexão será encerrada se permanecer acima do limite flexível por mais do que o tempo em segundos definido em `client-output-buffer-limit-normal-soft-seconds` ou se atingir o limite rígido;
- `client-output-buffer-limit-normal-soft-seconds`: tempo permitido para as ligações que excedem o `client-output-buffer-limit-normal-soft-limit`;
- `client-output-buffer-limit-normal-hard-limit`: uma conexão que atinge esse limite será imediatamente encerrada.

Além dos buffers de cliente regulares, as seguintes opções controlam o buffer para nós de réplica e clientes 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`;

Solução de problemas no lado do cliente para instâncias do Amazon EC2

A carga e a capacidade de resposta no lado do cliente também podem afetar as solicitações ao ElastiCache. Os limites de instância e sistema operacional do EC2 precisam ser cuidadosamente revisados ao solucionar problemas de conectividade intermitente ou tempo limite. Alguns pontos-chave a observar:

- CPU:

- Uso da CPU da instância do EC2: verifique se a CPU não está saturada ou perto de 100%. A análise histórica pode ser feita via CloudWatch, no entanto, tenha em mente que a granularidade dos pontos de dados é de 1 minuto (com monitoramento detalhado ativado) ou 5 minutos;
- Se estiver usando as [instâncias do EC2 com intermitência](#), certifique-se de que o saldo de crédito da CPU não tenha se esgotado. Essas informações estão disponíveis na métrica CPUCreditBalance do CloudWatch
- Períodos curtos de alto uso da CPU podem causar estouros de tempos limite sem refletir em 100% de utilização no CloudWatch. Tais casos exigem monitoramento em tempo real com ferramentas do sistema operacional como top, ps e mpstat.
- Rede
 - Verifique se o throughput de rede está abaixo de valores aceitáveis de acordo com os recursos da instância. Para obter mais informações, consulte [Tipos de instâncias do Amazon EC2](#)
 - Em instâncias com o Driver de rede aprimorado ena, verifique as [estatísticas do ena](#) sobre tempo limite ou limites excedidos. As estatísticas a seguir são úteis para confirmar a saturação de limites de rede:
 - bw_in_allowance_exceeded / bw_out_allowance_exceeded: número de pacotes modelados devido ao throughput excessivo de entrada ou saída;
 - conntrack_allowance_exceeded: número de pacotes descartados devido a [limites de monitoramento de conexão](#) de grupos de segurança. Novas conexões falharão quando esse limite estiver saturado;
 - linklocal_allowance_exceeded: número de pacotes descartados devido a solicitações excessivas de metadados de instâncias, NTP via VPC DNS. O limite é de 1024 pacotes por segundo para todos os serviços;
 - pps_allowance_exceeded: número de pacotes descartados devido à proporção excessiva de pacotes por segundo. O limite de PPS pode ser atingido quando o tráfego de rede consistir em milhares ou milhões de solicitações muito pequenas por segundo. O tráfego do ElastiCache pode ser otimizado para fazer melhor uso de pacotes de rede por meio de pipelines ou comandos que façam várias operações ao mesmo tempo, como MGET em vez de GET.

Dissecação do tempo necessário para concluir uma única solicitação

- Na rede: Tcpdump e Wireshark (tshark na linha de comando) são ferramentas úteis para entender quanto tempo a solicitação levou para viajar pela rede, bater no mecanismo do

ElastiCache e obter um retorno. O exemplo a seguir destaca uma única solicitação criada com o seguinte comando:

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

Paralelamente ao comando acima, tcpdump estava em execução e retornou:

```
$ 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
```

A partir da saída acima, podemos confirmar que o handshake de três vias do TCP foi concluído em 222 microssegundos (918091 - 917869) e o comando ping foi enviado e retornado em 173 microssegundos (918295 - 918122).

Demorou 438 microssegundos (918307 - 917869) da solicitação ao fechamento da conexão. Esses resultados confirmam que os tempos de resposta da rede e do mecanismo são bons e que a investigação pode concentrar-se em outros componentes.

- No sistema operacional: `Strace` pode ajudar a identificar intervalos de tempo no nível do sistema operacional. A análise de aplicações reais seria muito mais extensa, e profilers ou depuradores especializados de aplicações são aconselhados. O exemplo a seguir mostra apenas se os componentes básicos do sistema operacional estão funcionando como esperado, caso contrário, investigações adicionais podem ser necessárias. Usando do mesmo comando `PING` do Redis OSS com `strace`, obtemos:

```
$ 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\notls20201224\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\notls20201224\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 +++
```

No exemplo acima, o comando levou um pouco mais de 54 milissegundos para ser concluído (752110 - 697712 = 54398 microssegundos).

Um tempo significativo, aproximadamente 20 ms, foi levado para instanciar nc e fazer a resolução do nome (de 697712 a 717890), depois disso, 2ms foram necessários para criar o soquete de TCP (745659 a 747858), e 0,4 ms (747858 a 748330) para enviar e receber a resposta para a solicitação.

Related Topics

- [the section called “Práticas recomendadas e estratégias de armazenamento em cache”](#)

Segurança no Amazon ElastiCache

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você contará com um data center e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao Amazon ElastiCache, consulte [Serviços da AWS no escopo por programa de conformidade](#).
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon ElastiCache. Os tópicos a seguir mostram como configurar o Amazon ElastiCache para atender aos seus objetivos de segurança e de conformidade. Saiba também como usar outros serviços da AWS que ajudam você a monitorar e proteger os recursos do Amazon ElastiCache.

Tópicos

- [Proteção de dados no Amazon ElastiCache](#)
- [Privacidade do tráfego entre redes](#)
- [Identity and Access Management para Amazon ElastiCache](#)
- [Validação de conformidade para a Amazon ElastiCache](#)
- [Resiliência na Amazon ElastiCache](#)
- [Segurança da infraestrutura no ElastiCache da AWS](#)
- [Atualizações de serviço em ElastiCache](#)
- [Vulnerabilidades e exposições comuns \(CVE\): vulnerabilidades de segurança abordadas em ElastiCache](#)

Proteção de dados no Amazon ElastiCache

O [modelo de responsabilidade compartilhada](#) da AWS se aplica à proteção de dados no AWS ElastiCache (ElastiCache). Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos serviços da AWS que você usa. Para obter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#).

Para fins de proteção de dados, recomendamos que você proteja as credenciais da conta da AWS e configure as contas individuais com o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use TLS para se comunicar com os recursos da AWS.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão nos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais armazenados no Amazon S3.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o ElastiCache ou outros serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Todos os dados inseridos no ElastiCache ou em outros serviços podem ser selecionados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

Tópicos

- [Segurança de dados no Amazon ElastiCache](#)

Segurança de dados no Amazon ElastiCache

Para ajudar a manter seus dados seguros, o Amazon ElastiCache e o Amazon EC2 fornecem mecanismos para proteger contra o acesso não autorizado aos seus dados no servidor.

O Amazon ElastiCache para Memcached oferece atributos de criptografia para dados em caches que estiverem executando as versões 1.6.12 ou posterior do Memcached.

O Amazon ElastiCache com Valkey e Redis OSS oferece atributos de criptografia para dados em caches que estiverem executando o Valkey 7.2 ou posterior e as versões 3.2.6 do Redis OSS (programadas para EOL, consulte [Cronograma de fim de vida útil das versões do Redis OSS](#)), 4.0.10 ou posteriores. O Amazon ElastiCache dá suporte à autenticação de usuários com o IAM ou o AUTH do Valkey e Redis OSS, além da autorização das operações de usuários usando o controle de acesso baseado em função (RBAC).

- A criptografia em trânsito criptografa seus dados sempre que eles estão se movendo de um lugar para outro, como entre os nós no seu cluster ou entre seu cache e o aplicativo.
- A criptografia em repouso criptografa dados no disco durante as operações de sincronização e backup.

O ElastiCache oferece suporte à autenticação de usuários usando o IAM e o comando AUTH do Valkey e Redis OSS, além de autorizar operações de usuários usando o controle de acesso baseado em função (RBAC).

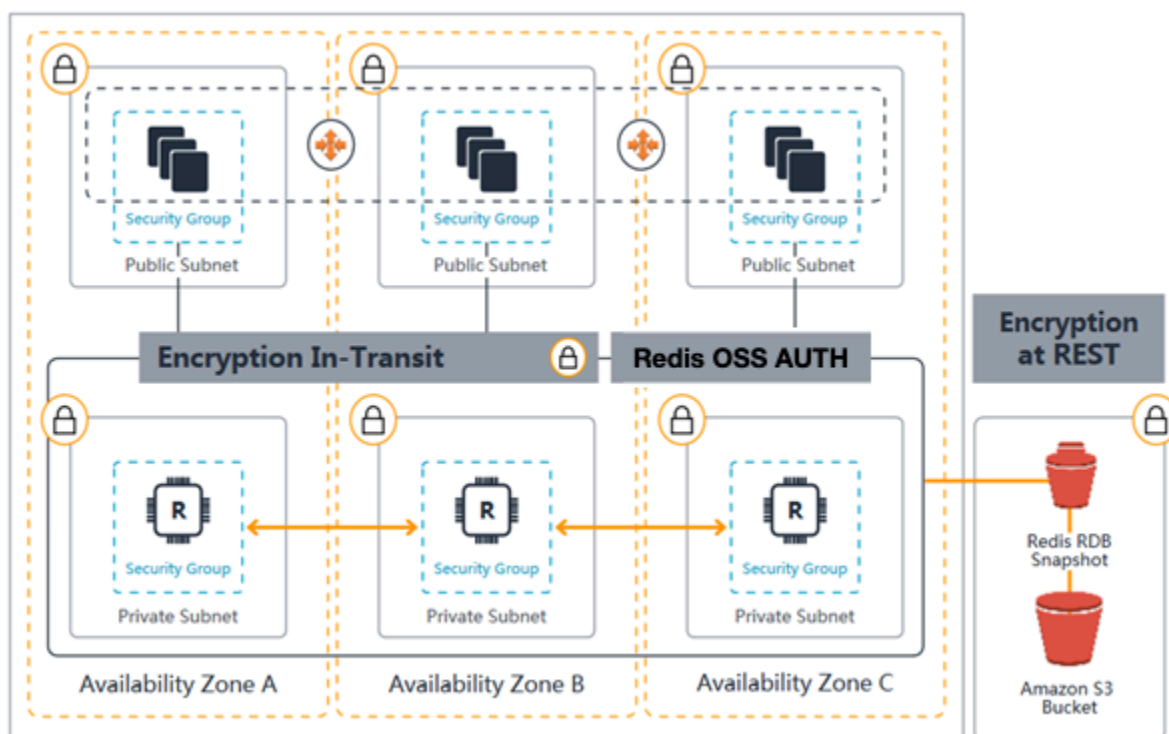


Diagrama de segurança do ElastiCache para Valkey e Redis OSS

Tópicos

- [ElastiCache criptografia em trânsito \(TLS\)](#)
- [Criptografia em repouso em ElastiCache](#)
- [Autenticação e autorização](#)

ElastiCache criptografia em trânsito (TLS)

Para ajudar a manter seus dados seguros, a Amazon ElastiCache e a Amazon EC2 fornecem mecanismos de proteção contra o acesso não autorizado aos seus dados no servidor. Ao fornecer o recurso de criptografia em trânsito, ElastiCache oferece uma ferramenta que você pode usar para ajudar a proteger seus dados quando eles são movidos de um local para outro.

Todos os caches de tecnologia sem servidor do Valkey ou Redis OSS têm criptografia em trânsito habilitada. Para clusters baseados em nós, você pode habilitar a criptografia em trânsito em um grupo de replicação configurando o parâmetro `TransitEncryptionEnabled` como `true` (CLI: `--transit-encryption-enabled`) ao criar o grupo de replicação. Você pode fazer isso se estiver criando o grupo de replicação usando a Console de gerenciamento da AWS, a AWS CLI, a ou a ElastiCache API.

Todos os caches sem servidor têm criptografia em trânsito habilitada. Para clusters baseados em nós, você pode habilitar a criptografia em trânsito em um cluster definindo o parâmetro como `TransitEncryptionEnabled` (`true` CLI: `--transit-encryption-enabled`) ao criar o cluster usando a operação `CreateCacheCluster` (CLI: `create-cache-cluster`).

Tópicos

- [Visão geral da criptografia em trânsito](#)
- [Condições de criptografia em trânsito \(Valkey e Redis OSS\)](#)
- [Condições de criptografia em trânsito \(Memcached\)](#)
- [Práticas recomendadas de criptografia em trânsito](#)
- [Outras opções para Valkey e Redis OSS](#)
- [Habilitação da criptografia em trânsito para o Memcached](#)
- [Habilitação da criptografia em trânsito](#)
- [Conexão ao ElastiCache \(Valkey\) ou ao Amazon ElastiCache para Redis OSS com criptografia em trânsito usando o valkey-cli](#)
- [Ativar criptografia em trânsito em um cluster Redis OSS baseado em nós usando Python](#)

- [Práticas recomendadas ao habilitar a criptografia em trânsito](#)
- [Conectar-se a nós habilitados com criptografia em trânsito usando o Openssl \(Memcached\)](#)
- [Criação de um cliente TLS Memcached usando Java](#)
- [Criação de um cliente TLS Memcached usando PHP](#)

Visão geral da criptografia em trânsito

A criptografia ElastiCache em trânsito da Amazon é um recurso que permite aumentar a segurança de seus dados nos pontos mais vulneráveis, quando estão em trânsito de um local para outro. Como a criptografia e descriptografia dos dados requerem processamento nos endpoints, a ativação da criptografia em trânsito pode ter impacto no desempenho. Compare seus dados com e sem criptografia em trânsito para determinar o impacto no desempenho para seus casos de uso.

ElastiCache a criptografia em trânsito implementa os seguintes recursos:

- Conexões de cliente criptografadas: as conexões do cliente com os nós de cache são criptografadas por TLS.
- Conexões de servidor criptografadas: os dados que se movem entre os nós em um cluster são criptografados.
- Autenticação do servidor: os clientes podem autenticar que estão conectados ao servidor certo.
- Autenticação do cliente: usando o atributo AUTH do Valkey ou do Redis OSS, o servidor pode autenticar os clientes.

Condições de criptografia em trânsito (Valkey e Redis OSS)

As seguintes restrições na criptografia ElastiCache em trânsito da Amazon devem ser lembradas ao planejar sua implementação de cluster baseado em nós:

- A criptografia em trânsito é aceita em grupos de replicação que executam o Valkey 7.2 e posteriores, e o Redis OSS versões 3.2.6, 4.0.10 e posteriores.
- A modificação da configuração de criptografia em trânsito para um cluster existente é aceita em grupos de replicação que executam o Valkey 7.2 e posterior, e o Redis OSS versão 7 e posterior.
- A criptografia em trânsito tem suporte somente em grupos de replicação em execução em uma Amazon VPC.
- A criptografia em trânsito não é aceita para grupos de replicação que executam os seguintes tipos de nó: M1, M2.

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- A criptografia em trânsito é ativada configurando explicitamente o parâmetro `TransitEncryptionEnabled` como `true`.
- Verifique se o cliente de armazenamento em cache dá suporte à conectividade TLS e se você a habilitou na configuração do cliente.
- A partir de 26 de janeiro de 2026, AWS atualizaremos a versão mínima suportada do TLS para 1.2 no ElastiCache Valkey versão 7.2 e superior e no Redis OSS versão 6 e ElastiCache superior. Os clientes devem atualizar o software-cliente antes dessa data. Essa atualização ajuda você a atender às necessidades regulatórias, de conformidade e de segurança.

Condições de criptografia em trânsito (Memcached)

As seguintes restrições na criptografia ElastiCache em trânsito da Amazon devem ser lembradas ao planejar sua implementação de cluster baseado em nós:

- A criptografia em trânsito é compatível com clusters executando as versões 1.6.12 e posteriores do Memcached.
- A criptografia em trânsito é compatível com as versões 1.2 e 1.3 do Transport Layer Security (TLS).
- A criptografia em trânsito é compatível somente em clusters em execução em uma Amazon VPC.
- A criptografia em trânsito não é aceita para grupos de replicação que executam os seguintes tipos de nó: M1, M2, M3, R3, T2.

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- A criptografia em trânsito é ativada configurando explicitamente o parâmetro `TransitEncryptionEnabled` como `true`.
- Só é possível ativar a criptografia em trânsito em um cluster ao criá-lo. Não é possível ativar e desativar a criptografia em trânsito modificando um cluster.
- Verifique se o cliente de armazenamento em cache dá suporte à conectividade TLS e se você a habilitou na configuração do cliente.

Práticas recomendadas de criptografia em trânsito

- Como a criptografia e descriptografia dos dados requerem processamento nos endpoints, a implementação da criptografia em trânsito pode reduzir o desempenho. Compare seus dados

com criptografia em trânsito e sem criptografia para determinar o impacto no desempenho da sua implementação.

- Como criar novas conexões pode ser caro, é possível reduzir o impacto na performance da criptografia em trânsito persistindo suas conexões TLS.

Outras opções para Valkey e Redis OSS

Para obter mais informações sobre as opções disponíveis para Valkey e Redis OSS, consulte os links a seguir.

- [Criptografia em repouso em ElastiCache](#)
- [Autenticar com o comando AUTH do Valkey e Redis OSS](#)
- [Regras de controle de acesso com base em função \(RBAC\)](#)
- [Amazon VPCs e ElastiCache segurança](#)
- [Identity and Access Management para Amazon ElastiCache](#)

Habilitação da criptografia em trânsito para o Memcached

Para ativar a criptografia em trânsito ao criar um cluster Memcached usando o AWS Management Console, escolha as seguintes opções:

- Escolha o Memcached como seu mecanismo.
- Escolha a versão 1.6.12 ou posterior do mecanismo.
- Em Encryption in transit (Criptografia em trânsito), escolha Enable (Habilitar).

Para o step-by-step processo, consulte [Criação de um cluster do Valkey ou Redis OSS](#).

Habilitação da criptografia em trânsito

Todos os caches sem servidor têm criptografia em trânsito habilitada. Em um cluster baseado em nós, você pode ativar a criptografia em trânsito usando a Console de gerenciamento da AWS a AWS CLI ou a API. ElastiCache

Habilitando a criptografia em trânsito usando o Console de gerenciamento da AWS

Habilitando a criptografia em trânsito para um novo cluster baseado em nós usando o Console de gerenciamento da AWS

Ao projetar o próprio cluster, as configurações “Dev/Teste” e “Produção” com o método “Criação fácil” têm a criptografia em trânsito habilitada. Ao escolher a configuração por conta própria, faça as seguintes seleções:

- Escolha a versão 3.2.6, 4.0.10 ou posterior do mecanismo.
- Clique na caixa de seleção ao lado de Habilitar para a opção Criptografia em trânsito.

Para o step-by-step processo, consulte o seguinte:

- [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#)
- [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)

Habilitando a criptografia em trânsito para um cluster existente baseado em nós usando o Console de gerenciamento da AWS

Habilitar a criptografia em trânsito é um processo de duas etapas para o qual você deve primeiro definir o modo de criptografia de em trânsito como `preferred`. Esse modo permite que seus clientes Valkey ou Redis OSS se conectem usando conexões criptografadas e não criptografadas. Depois de migrar todos os seus clientes Valkey ou Redis OSS para usar conexões criptografadas, você pode modificar a configuração do cluster para definir o modo de criptografia em trânsito como `required`. Se o modo de criptografia em trânsito for definido como `required`, todas as conexões não criptografadas serão eliminadas e somente conexões criptografadas serão permitidas.

Defina seu modo de criptografia em trânsito como Preferencial

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Escolha caches Valkey ou caches Redis OSS nos ElastiCache Recursos listados no painel de navegação, presentes à esquerda.
3. Escolha o cache que você deseja atualizar.
4. Escolha o menu suspenso Actions (Ações) e escolha Modify (Modificar).

5. Escolha Enable (Habilitar) em Encryption in transit (Criptografia em trânsito) na seção Security (Segurança).
6. Escolha Preferred (Preferencial) como Transit encryption mode (Modo de criptografia em trânsito).
7. Escolha Preview changes (Visualizar alterações) e salve suas alterações.

Depois de migrar todos os seus clientes Valkey ou Redis OSS para usar conexões criptografadas:

Defina seu Modo de criptografia em trânsito como Obrigatório

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Escolha caches Valkey ou caches Redis OSS nos ElastiCache Recursos listados no painel de navegação, presentes à esquerda.
3. Escolha o cache que você deseja atualizar.
4. Escolha o menu suspenso Actions (Ações) e escolha Modify (Modificar).
5. Escolha Required (Obrigatório) como Transit encryption mode (Modo de criptografia em trânsito), na seção Security (Segurança).
6. Escolha Preview changes (Visualizar alterações) e salve suas alterações.

Habilitando a criptografia em trânsito usando o AWS CLI

Para habilitar a criptografia em trânsito ao criar um grupo de replicação do Valkey ou do Redis OSS usando a AWS CLI, use o parâmetro `transit-encryption-enabled`.

Habilitação da criptografia em trânsito em um cluster baseado em nós para Valkey ou Redis OSS (modo cluster desabilitado) (CLI)

Use a AWS CLI operação `create-replication-group` e os parâmetros a seguir para criar um grupo de replicação Valkey ou Redis OSS com réplicas que tenham a criptografia em trânsito ativada:

Principais parâmetros:

- **--engine:** precisa ser `valkey` ou `redis`.
- **--engine-version:** se o mecanismo for Redis OSS, é preciso que seja 3.2.6, 4.0.10 ou posterior.

- **--transit-encryption-enabled**—Obrigatório. Se você habilitar a criptografia em trânsito, também deverá fornecer um valor para o parâmetro `--cache-subnet-group`.
- **--num-cache-clusters**: deve ser pelo menos 1. O valor máximo para esse parâmetro é de seis.

Para saber mais, consulte:

- [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) do início \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitação da criptografia em trânsito em um cluster baseado em nós para Valkey ou Redis OSS (modo cluster habilitado) (CLI)

Use a AWS CLI operação `create-replication-group` e os parâmetros a seguir para criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) que tenha a criptografia em trânsito ativada:

Principais parâmetros:

- **--engine**: precisa ser `valkey` ou `redis`.
- **--engine-version**: se o mecanismo for Redis OSS, é preciso que seja 3.2.6, 4.0.10 ou posterior.
- **--transit-encryption-enabled**: obrigatório. Se você habilitar a criptografia em trânsito, também deverá fornecer um valor para o parâmetro `--cache-subnet-group`.
- Use um dos seguintes conjuntos de parâmetros para especificar a configuração dos grupos de nó do grupo de replicação:
 - **--num-node-groups**: especifica o número de grupos de fragmentos (grupos de nós) deste grupo de replicação. O valor máximo desse parâmetro é de 500.
 - **--replicas-per-node-group**: especifica o número de nós de réplica em cada grupo de nós. O valor especificado aqui é aplicado a todos os fragmentos neste grupo de replicação. O valor máximo desse parâmetro é de 5.
 - **--node-group-configuration**: especifica a configuração de cada fragmento de forma independente.

Para saber mais, consulte:

- [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster habilitado\) do início \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitação da criptografia em trânsito em um cluster existente usando o AWS CLI

Habilitar a criptografia em trânsito é um processo de duas etapas para o qual você deve primeiro definir o modo de criptografia de em trânsito como `preferred`. Esse modo permite que seus clientes Valkey ou Redis OSS se conectem usando conexões criptografadas e não criptografadas. Depois de migrar todos os seus clientes Valkey ou Redis OSS para usar conexões criptografadas, você pode modificar a configuração do cluster para definir o modo de criptografia em trânsito como `required`. Se o modo de criptografia em trânsito for definido como `required`, todas as conexões não criptografadas serão eliminadas e somente conexões criptografadas serão permitidas.

Use a AWS CLI operação `modify-replication-group` e os parâmetros a seguir para atualizar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) que tenha a criptografia em trânsito desativada.

Como habilitar a criptografia em trânsito

1. `transit-encryption-mode` Defina como `preferred`, usando os seguintes parâmetros
 - **`--transit-encryption-enabled`**—Obrigatório.
 - **`--transit-encryption-mode`**: deve ser definido como `preferred`.
2. `transit-encryption-mode` Defina como `required`, usando os seguintes parâmetros:
 - **`--transit-encryption-enabled`**—Obrigatório.
 - **`--transit-encryption-mode`**: deve ser definido como `required`.

Conexão ao ElastiCache (Valkey) ou ao Amazon ElastiCache para Redis OSS com criptografia em trânsito usando o `valkey-cli`

Para acessar dados de caches do ElastiCache para Redis OSS habilitados com criptografia em trânsito, você usa clientes que trabalhem com Secure Socket Layer (SSL). Você também pode usar o `valkey-cli` com TLS/SSL no Amazon Linux e no Amazon Linux 2. Se o cliente não oferecer suporte

ao TLS, você poderá usar o comando `stunnel` no host do cliente a fim de criar um túnel SSL para os nós do Redis OSS.

Conexão criptografada com Linux

Para usar o `valkey-cli` para se conectar a um cluster do Valkey ou Redis OSS habilitado com criptografia em trânsito no Amazon Linux 2 ou Amazon Linux, siga estas etapas.

1. Baixe e compile o utilitário `valkey-cli`. Esse utilitário está incluído na distribuição do software Valkey.
2. No prompt de comando da sua instância do EC2, digite os comandos apropriados para a versão do Linux que você está usando.

Amazon Linux 2

Se estiver usando o Amazon Linux 2, digite o seguinte:

```
sudo yum -y install openssl-devel gcc
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean
make valkey-cli BUILD_TLS=yes
sudo install -m 755 src/valkey-cli /usr/local/bin/
```

Amazon Linux

Se estiver usando o Amazon Linux, digite o seguinte:

```
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel clang wget
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make valkey-cli CC=clang BUILD_TLS=yes
sudo install -m 755 src/valkey-cli /usr/local/bin/
```

No Amazon Linux, também pode ser necessário executar as seguintes etapas adicionais:

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

```
sudo make install
```

3. Depois de baixar e instalar o utilitário valkey-cli, recomendamos executar o comando opcional `make-test`.
4. Para se conectar a um cluster com criptografia e autenticação habilitadas, digite este comando:

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

Note

Se você instalar o redis6 no Amazon Linux 2023, agora poderá usar o comando `redis6-cli` em vez de `valkey-cli`:

```
redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
```

Conexão criptografada com stunnel

Para usar o valkey-cli para se conectar a um cluster do Redis OSS habilitado com criptografia em trânsito usando túnel, siga as etapas a seguir.

1. Use o SSH para se conectar ao seu cliente e instalar o stunnel.

```
sudo yum install stunnel
```

2. Execute o seguinte comando para criar e editar o arquivo `'/etc/stunnel/valkey-cli.conf'` simultaneamente para adicionar um endpoint de cluster do ElastiCache para Redis OSS a um ou mais parâmetros de conexão, usando a saída fornecida abaixo como modelo.

```
vi /etc/stunnel/valkey-cli.conf
```

```
fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
```

```
options = NO_SSLv3
[valkey-cli]
  client = yes
  accept = 127.0.0.1:6379
  connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[valkey-cli-replica]
  client = yes
  accept = 127.0.0.1:6380
  connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379
```

Neste exemplo, o arquivo de configuração tem duas conexões, `valkey-cli` e `valkey-cli-replica`. Os parâmetros são definidos como a seguir:

- `client` (cliente) é definido com `sim` para especificar que essa instância de stunnel é um cliente.
- `accept` (aceitar) é definido com o IP do cliente. Neste exemplo, o primário é definido com o Redis OSS padrão 127.0.0.1 na porta 6379. A réplica deve chamar uma porta diferente e ser definida como 6380. Você pode usar portas efêmeras 1024-65535. Para obter mais informações, consulte [Portas efêmeras](#) no Guia do usuário da Amazon VPC.
- `connect` é definido como o endpoint do servidor Redis OSS. Para obter mais informações, consulte [Encontrar endpoints de conexão no ElastiCache](#).

3. Início stunnel.

```
sudo stunnel /etc/stunnel/valkey-cli.conf
```

Use o comando `netstat` para confirmar que os túneis começaram a funcionar.

```
sudo netstat -tulnp | grep -i stunnel

tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTEN
3189/stunnel
tcp        0      0 127.0.0.1:6380          0.0.0.0:*               LISTEN
3189/stunnel
```

4. Conecte-se ao nó criptografado do Redis OSS usando o endpoint local do túnel.

- Se nenhuma senha AUTH foi usada durante a criação do cluster do ElastiCache para Redis OSS, este exemplo usa o `valkey-cli` para se conectar ao servidor ElastiCache para Redis OSS usando o caminho completo para o `valkey-cli`, no Amazon Linux:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
```

Se a senha AUTH foi usada durante a criação do cluster do Redis OSS, este exemplo usa o valkey-cli para se conectar ao servidor Redis OSS usando o caminho completo para o valkey-cli, no Amazon Linux:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

OU

- Altere o diretório para redis-7.2.5 e faça o seguinte:

Se nenhuma senha AUTH foi usada durante a criação do cluster do ElastiCache para Redis OSS, este exemplo usa o valkey-cli para se conectar ao servidor ElastiCache para Redis OSS usando o caminho completo para o valkey-cli, no Amazon Linux:

```
src/valkey-cli -h localhost -p 6379
```

Se a senha AUTH foi usada durante a criação do cluster do Redis OSS, este exemplo usa o valkey-cli para se conectar ao servidor Valkey ou Redis OSS usando o caminho completo para o valkey-cli, no Amazon Linux:

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

Este exemplo usa o Telnet para se conectar ao servidor Valkey ou Redis OSS.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
```



```
bar
```

5. Para encerrar e fechar os túneis SSL, `kill` o processo de `stunnel`.

```
sudo kill stunnel
```

Ativar criptografia em trânsito em um cluster Redis OSS baseado em nós usando Python

O guia a seguir demonstrará como ativar a criptografia em trânsito em um cluster do Redis OSS 7.0 que foi originalmente criado com a criptografia em trânsito desativada. Os clientes TCP e TLS continuarão se comunicando com o cluster durante esse processo sem tempo de inatividade.

O Boto3 obterá as credenciais necessárias

(`aws_access_key_id`, `aws_secret_access_key` e `aws_session_token`) das variáveis de ambiente. Essas credenciais serão coladas com antecedência no mesmo terminal `bash` em que executaremos o `python3` para processar o código Python mostrado nesse guia. O código no exemplo abaixo foi processado a partir de uma EC2 instância que foi executada na mesma VPC que será usada para criar o ElastiCache Redis OSS Cluster nela.

Note

- Os exemplos a seguir usam o SDK `boto3` para operações de ElastiCache gerenciamento (criação de cluster ou usuário) e `redis-py-cluster` `redis-py/` para tratamento de dados.
- Você deve usar pelo menos a versão `boto3` (`=~`) 1.26.39 para usar a migração TLS on-line com a API de modificação de cluster.
- ElastiCache oferece suporte à migração de TLS on-line somente para clusters com Valkey versão 7.2 e superior ou Redis OSS versão 7.0 ou superior. Portanto, se você tiver um cluster executando uma versão do Redis OSS anterior à 7.0, será preciso atualizar a versão do Redis OSS do seu cluster. Para obter mais informações sobre as diferenças entre as versões, consulte [Principais diferenças de comportamento e compatibilidade do mecanismo da versão com o Redis OSS](#).

Tópicos

- [Defina as constantes de string que iniciarão o ElastiCache Valkey ou o Redis OSS Cluster](#)
- [Defina as classes para a configuração do cluster](#)

- [Defina uma classe que representará o próprio cluster](#)
- [\(Opcional\) Crie uma classe wrapper para demonstrar a conexão do cliente com o cluster do Valkey ou Redis OSS](#)
- [Crie a função principal que demonstra o processo de alteração da configuração de criptografia em trânsito](#)

Defina as constantes de string que iniciarão o ElastiCache Valkey ou o Redis OSS Cluster

Primeiro, vamos definir algumas constantes de string simples do Python que conterão os nomes das AWS entidades necessárias para criar o ElastiCache clustersecurity-group, comoCache Subnet group, e a default parameter group. Todas essas AWS entidades devem ser criadas com antecedência em sua AWS conta na região que você deseja usar.

```
#Constants definitions
SECURITY_GROUP = "sg-0492aa0a29c558427"
CLUSTER_DESCRIPTION = "This cluster has been launched as part of the online TLS
migration user guide"
EC_SUBNET_GROUP = "client-testing"
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED = "default.redis7.cluster.on"
```

Defina as classes para a configuração do cluster

Agora, vamos definir algumas classes simples de Python que representarão a configuração de um cluster, que conterá metadados sobre o cluster, como a versão do Valkey ou do Redis OSS, o tipo de instância e se a criptografia em trânsito (TLS) está habilitada ou desabilitada.

```
#Class definitions

class Config:
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
    ):
        self.instance_type = instance_type
        self.version = version
        self.multi_az = multi_az
```

```
self.TLS = TLS
self.name = name or f"tls-test"

def create_base_launch_request(self):
    return {
        "ReplicationGroupId": self.name,
        "TransitEncryptionEnabled": self.TLS,
        "MultiAZEnabled": self.multi_az,
        "CacheNodeType": self.instance_type,
        "Engine": "redis",
        "EngineVersion": self.version,
        "CacheSubnetGroupName": EC_SUBNET_GROUP ,
        "CacheParameterGroupName":
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED ,
        "ReplicationGroupDescription": CLUSTER_DESCRIPTION,
        "SecurityGroupIds": [SECURITY_GROUP],
    }

class ConfigCME(Config):
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
        num_shards: int = 2,
        num_replicas_per_shard: int = 1,
    ):
        super().__init__(instance_type, version, multi_az, TLS, name)
        self.num_shards = num_shards
        self.num_replicas_per_shard = num_replicas_per_shard

    def create_launch_request(self) -> dict:
        launch_request = self.create_base_launch_request()
        launch_request["NumNodeGroups"] = self.num_shards
        launch_request["ReplicasPerNodeGroup"] = self.num_replicas_per_shard
        return launch_request
```

Defina uma classe que representará o próprio cluster

Agora, vamos definir algumas classes simples de Python que representarão o próprio ElastiCache Valkey ou Redis OSS Cluster. Essa classe terá um campo de cliente que conterá um cliente boto3 para operações ElastiCache de gerenciamento, como criar o cluster e consultar a API. ElastiCache

```
import botocore.config
import boto3

# Create boto3 client
def init_client(region: str = "us-east-1"):
    config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
    init_request = dict()
    init_request["config"] = config
    init_request["service_name"] = "elasticache"
    init_request["region_name"] = region
    return boto3.client(**init_request)

class ElastiCacheClusterBase:
    def __init__(self, name: str):
        self.name = name
        self.elasticache_client = init_client()

    def get_first_replication_group(self):
        return self.elasticache_client.describe_replication_groups(
            ReplicationGroupId=self.name
        )["ReplicationGroups"][0]

    def get_status(self) -> str:
        return self.get_first_replication_group()["Status"]

    def get_transit_encryption_enabled(self) -> bool:
        return self.get_first_replication_group()["TransitEncryptionEnabled"]

    def is_available(self) -> bool:
        return self.get_status() == "available"

    def is_modifying(self) -> bool:
        return self.get_status() == "modifying"

    def wait_for_available(self):
        while True:
```

```
        if self.is_available():
            break
        else:
            time.sleep(5)

def wait_for_modifying(self):
    while True:
        if self.is_modifying():
            break
        else:
            time.sleep(5)

def delete_cluster(self) -> bool:
    self.elasticache_client.delete_replication_group(
        ReplicationGroupId=self.name, RetainPrimaryCluster=False
    )

def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
    # generate api call to migrate the cluster to TLS preferred or to TLS required
    self.elasticache_client.modify_replication_group(
        ReplicationGroupId=self.name,
        TransitEncryptionMode=new_transit_encryption_mode,
        TransitEncryptionEnabled=True,
        ApplyImmediately=True,
    )
    self.wait_for_modifying()

class ElastiCacheClusterCME(ElastiCacheClusterBase):
    def __init__(self, name: str):
        super().__init__(name)

    @classmethod
    def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
        config = config or ConfigCME()
        print(config)
        new_cluster = ElastiCacheClusterCME(config.name)
        launch_request = config.create_launch_request()
        new_cluster.elasticache_client.create_replication_group(**launch_request)
        new_cluster.wait_for_available()
        return new_cluster

    def get_configuration_endpoint(self) -> str:
        return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]
```

```
#Since the code can throw exceptions, we define this class to make the code more
readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
    def __init__(self, config: ConfigCME = None):
        self.config = config or ConfigCME()

    def __enter__(self) -> ElastiCacheClusterCME:
        self.cluster = ElastiCacheClusterCME.launch(self.config)
        return self.cluster

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.cluster.delete_cluster()
```

(Opcional) Crie uma classe wrapper para demonstrar a conexão do cliente com o cluster do Valkey ou Redis OSS

Agora, vamos criar uma classe de wrapper para o cliente `redis-py-cluster`. Essa classe de wrapper será compatível com o pré-preenchimento do cluster com algumas chaves e, em seguida, com a execução de comandos `get` aleatórios repetidos.

Note

Essa é uma etapa opcional, mas simplifica o código da função principal que vem em uma etapa posterior.

```
import redis
import random
from time import perf_counter_ns, time

class DowntimeTestClient:
    def __init__(self, client):
        self.client = client

        # num of keys prefilled
        self.prefilled = 0
        # percent of get above prefilled
        self.percent_get_above_prefilled = 10 # nil result expected when get hit above
        prefilled
        # total downtime in nano seconds
```

```

        self.downtime_ns = 0
        # num of success and fail operations
        self.success_ops = 0
        self.fail_ops = 0
        self.connection_errors = 0
        self.timeout_errors = 0

def replace_client(self, client):
    self.client = client

def prefill_data(self, timelimit_sec=60):
    end_time = time() + timelimit_sec
    while time() < end_time:
        self.client.set(self.prefilled, self.prefilled)
        self.prefilled += 1

# unsuccessful operations throw exceptions
def _exec(self, func):
    try:
        start_ns = perf_counter_ns()
        func()
        self.success_ops += 1
        elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
        # upon succesful execution of func
        # reset random_key to None so that the next command
        # will use a new random key
        self.random_key = None

    except Exception as e:
        elapsed_ns = perf_counter_ns() - start_ns
        self.downtime_ns += elapsed_ns
        # in case of failure- increment the relevant counters so that we will keep
track
        # of how many connection issues we had while trying to communicate with
        # the cluster.
        self.fail_ops += 1
        if e.__class__ is redis.exceptions.ConnectionError:
            self.connection_errors += 1
        if e.__class__ is redis.exceptions.TimeoutError:
            self.timeout_errors += 1

def _repeat_exec(self, func, seconds):
    end_time = time() + seconds

```

```
while time() < end_time:
    self._exec(func)

def _new_random_key_if_needed(self, percent_above_prefilled):
    if self.random_key is None:
        max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
        return random.randint(0, max)
    return self.random_key

def _random_get(self):
    key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
    result = self.client.get(key)
    # we know the key was set for sure only in the case key < self.prefilled
    if key < self.prefilled:
        assert result.decode("UTF-8") == str(key)

def repeat_get(self, seconds=60):
    self._repeat_exec(self._random_get, seconds)

def get_downtime_ms(self) -> int:
    return self.downtime_ns // 10 ** 6

def do_get_until(self, cond_check):
    while not cond_check():
        self.repeat_get()
    # do one more get cycle once condition is met
    self.repeat_get()
```

Crie a função principal que demonstra o processo de alteração da configuração de criptografia em trânsito

Agora, vamos definir a função principal, que fará o seguinte:

1. Crie o cluster usando o cliente boto3 ElastiCache .
2. Inicialize o cliente `redis-py-cluster` que se conectará ao cluster com uma conexão TCP clara sem TLS.
3. O cliente `redis-py-cluster` preenche o cluster com alguns dados.
4. O cliente boto3 acionará a migração de TLS de não TLS para TLS preferencial.

5. Enquanto o cluster estiver sendo migrado para o TLS Preferred, o cliente TCP `redis-py-cluster` enviará operações `get` repetidas para o cluster até que a migração seja concluída.
6. Após a conclusão da migração para o TLS Preferred, afirmaremos que o cluster é compatível com a criptografia em trânsito. Depois, criaremos um cliente `redis-py-cluster` que se conectará ao cluster com TLS.
7. Enviaremos alguns comandos `get` usando o novo cliente TLS e o antigo cliente TCP.
8. O cliente `boto3` acionará a migração de TLS de TLS Preferred para TLS obrigatório.
9. Enquanto o cluster estiver sendo migrado para o TLS obrigatório, o cliente `redis-py-cluster` TLS enviará `get` operações repetidas para o cluster até que a migração seja concluída.

```
import redis

def init_cluster_client(
    cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
DowntimeTestClient:
    # we must use for the host name the cluster configuration endpoint.
    redis_client = redis.RedisCluster(
        host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
        socket_connect_timeout=0.1
    )
    test_client = DowntimeTestClient(redis_client)
    if prefill_data:
        test_client.prefill_data()
    return test_client

if __name__ == '__main__':
    config = ConfigCME(TLS=False, instance_type="cache.m5.large")

    with ElastiCacheCMEManager(config) as cluster:
        # create a client that will connect to the cluster with clear tcp connection
        test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

        # migrate the cluster to TLS Preferred
        cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

        # do repeated get commands until the cluster finishes the migration to TLS
        Preferred
        test_client_tcp.do_get_until(cluster.is_available)
```

```
# verify that in transit encryption is enabled so that clients will be able to
connect to the cluster with TLS
assert cluster.get_transit_encryption_enabled() == True

# create a client that will connect to the cluster with TLS connection.
# we must first make sure that the cluster indeed supports TLS
test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

# by doing get commands with the tcp client for 60 more seconds
# we can verify that the existing tcp connection to the cluster still works
test_client_tcp.repeat_get(seconds=60)

# do get commands with the new TLS client for 60 more seconds
test_client_tcp.repeat_get(seconds=60)

# migrate the cluster to TLS required
cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

# from this point the tcp clients will be disconnected and we must not use them
anymore.
# do get commands with the TLS client until the cluster finishes migration to
TLS required mode.
test_client_tls.do_get_until(cluster.is_available)
```

Práticas recomendadas ao habilitar a criptografia em trânsito

Antes de ativar a criptografia em trânsito: verifique se você tem o tratamento adequado dos registros DNS

Note

Estamos alterando e excluindo endpoints antigos durante esse processo. O uso incorreto dos endpoints pode fazer com que o cliente Valkey ou Redis OSS use endpoints antigos e excluídos, o que impedirá que ele se conecte ao cluster.

Enquanto o cluster está sendo migrado de não TLS para TLS preferencial, o registro DNS do endpoint de configuração de cluster antigo é mantido e os novos registros DNS do endpoint de configuração de cluster são gerados em um formato diferente. Os clusters habilitados para TLS usam um formato de registros DNS diferente dos clusters desabilitados para TLS. O ElastiCache manterá os dois registros DNS quando um cluster for configurado no `encryption mode: Preferred` para

que aplicações e outros clientes Valkey ou Redis OSS possam alternar entre eles. As seguintes alterações nos registros DNS ocorrem durante o processo de migração do TLS:

Descrição das alterações nos registros DNS que ocorrem ao ativar a criptografia em trânsito

Para clusters CME

Quando um cluster é definido como “modo de criptografia em trânsito: preferencial”:

- O endpoint do cluster de configuração original para o cluster não TLS permanecerá ativo. Não haverá tempo de inatividade quando o cluster for reconfigurado do modo de criptografia TLS 'nenhum' para 'preferencial'.
- Novos endpoints TLS Valkey ou Redis OSS serão gerados quando o cluster for definido no modo de TLS preferencial. Esses novos endpoints terão os mesmos IPs dos antigos (não TLS).
- O novo endpoint de configuração do TLS Valkey ou Redis OSS será exposto no console do ElastiCache e na resposta à API `describe-replication-group`.

Quando um cluster é definido como “modo de criptografia em trânsito: obrigatório”:

- Os endpoints habilitados para não TLS antigos serão excluídos. Não haverá tempo de inatividade dos endpoints do cluster TLS.
- Você pode recuperar um novo `cluster-configuration-endpoint` do console do ElastiCache ou da API `describe-replication-group`.

Para clusters CMD com Failover automático ativado ou Failover automático desativado

Quando um grupo de replicação é definido como “modo de criptografia em trânsito: preferencial”:

- O endpoint primário original e o endpoint do leitor para um cluster habilitado para não TLS permanecerão ativos.
- Os novos endpoints TLS primários e do leitor serão gerados quando o cluster for definido no modo de TLS Preferred. Esses novos endpoints terão os mesmos IPs dos antigos (não TLS).
- Os novos endpoints primário e de leitor serão expostos no console do ElastiCache e na resposta à API `describe-replication-group`.

Quando o grupo de replicação é definido como “modo de criptografia em trânsito: obrigatório”:

- Os endpoints primários e de leitura não TLS antigos serão excluídos. Não haverá tempo de inatividade dos endpoints do cluster TLS.
- Você pode recuperar os novos endpoints primário e do leitor no console do ElastiCache ou na API `describe-replication-group`.

Uso sugerido dos registros DNS

Para clusters CME

- Use o endpoint de configuração do cluster em vez dos registros DNS por nó no código do seu aplicativo. Não é recomendável usar nomes DNS por nó diretamente, pois eles mudarão durante a migração e o código do aplicativo interromperá a conexão com o cluster.
- Não codifique um endpoint de configuração de cluster em seu aplicativo, pois ele mudará durante esse processo.
- Ter o endpoint de configuração do cluster codificado em seu aplicativo é uma prática ruim, pois ele pode ser alterado durante esse processo. Depois que a criptografia em trânsito for concluída, consulte o endpoint de configuração do cluster com a API `describe-replication-group` [conforme demonstrado acima (em negrito)] e use o DNS que você obtiver em resposta a partir desse momento.

Para clusters CMD com Failover Automático ativado

- Use o endpoint primário e o endpoint do leitor em vez dos nomes DNS por nó no código do seu aplicativo, pois os nomes DNS por nó antigos são excluídos e os novos são gerados ao migrar o cluster do não TLS para o TLS preferencial. Não é recomendável usar nomes DNS por nó diretamente, pois você pode adicionar réplicas ao seu cluster no futuro. Além disso, quando o Failover Automático está ativado, as funções do cluster primário e das réplicas são alteradas automaticamente pelo serviço ElastiCache. Sugere-se usar o endpoint primário e o endpoint do leitor para ajudar você a acompanhar essas alterações. Por fim, usar o endpoint do leitor ajudará você a distribuir as leituras das réplicas igualmente entre as réplicas no cluster.
- Ter o endpoint primário e o endpoint do leitor codificados em seu aplicativo é uma prática ruim, pois isso pode ser alterado durante o processo de migração do TLS. Depois que a alteração da migração para o TLS preferencial for concluída, consulte o endpoint primário e o endpoint do leitor com a API `describe-replication-group` e use o DNS que você obtiver em resposta a partir desse momento. Dessa forma, você poderá acompanhar as mudanças nos endpoints de forma dinâmica.

Para cluster CMD com Failover automático ativado

- Use o endpoint primário e o endpoint do leitor em vez dos nomes DNS por nó no código do seu aplicativo. Quando o Failover automático está desativado, o ajuste de escala, a aplicação de patches, o failover e outros procedimentos que são gerenciados automaticamente pelo serviço ElastiCache quando o Failover automático está ativado são feitos por você. Isso facilita seu acompanhamento manual dos diferentes endpoints. Como os nomes DNS por nó antigos são excluídos e os novos são gerados ao migrar o cluster de não TLS para TLS preferencial, não use os nomes DNS por nó diretamente. Isso é obrigatório para que os clientes possam se conectar ao cluster durante a migração para TLS. Além disso, você se beneficiará de distribuir uniformemente as leituras entre as réplicas ao usar o endpoint do leitor e acompanhar os registros de DNS ao adicionar ou excluir réplicas do cluster.
- Ter o endpoint de configuração do cluster codificado em seu aplicativo é uma prática ruim, pois ele pode ser alterado durante o processo de migração TLS.

Durante a criptografia em trânsito: preste atenção quando o processo de migração terminar

A alteração do modo de criptografia em trânsito não é imediata e pode levar algum tempo. Isso é especialmente verdade para clusters grandes. Somente quando o cluster conclui a migração para o TLS preferencial é que ele pode aceitar e servir conexões TCP e TLS. Portanto, você não deve criar clientes que tentarão estabelecer conexões TLS com o cluster até que a criptografia em trânsito seja concluída.

Há várias maneiras de ser notificado quando a criptografia em trânsito é concluída com êxito ou falha: (não mostrado no exemplo de código acima):

- Usar o serviço SNS para receber uma notificação quando a criptografia for concluída
- Usar a API `describe-events` que emitirá um evento quando a criptografia for concluída
- Mensagem no console do ElastiCache informando que a criptografia foi concluída

Você também pode implementar a lógica em seu aplicativo para saber se a criptografia foi concluída. No exemplo acima, vimos várias maneiras de garantir que o cluster conclua a migração:

- Esperar até o início do processo de migração (o status do cluster muda para “modificando”) e esperar até que a modificação seja concluída (o status do cluster volta para “disponível”)
- Verificar se `transit_encryption_enabled` do cluster está definido como Verdadeiro consultando a API `describe-replication-group`.

Depois de ativar a criptografia em trânsito: verifique se os clientes que você usa estão configurados corretamente

Enquanto o cluster estiver no modo de TLS preferencial, seu aplicativo deve abrir conexões TLS com o cluster e usar somente essas conexões. Dessa forma, seu aplicativo não sofrerá tempo de inatividade ao ativar a criptografia em trânsito. Você pode garantir que não haja conexões TCP mais claras com o mecanismo Valkey ou Redis OSS usando o comando `info` na seção SSL.

```
# SSL
ssl_enabled:yes
ssl_current_certificate_not_before_date:Mar 20 23:27:07 2017 GMT
ssl_current_certificate_not_after_date:Feb 24 23:27:07 2117 GMT
ssl_current_certificate_serial:D8C7DEA91E684163
tls_mode_connected_tcp_clients:0    (should be zero)
tls_mode_connected_tls_clients:100
```

Conectar-se a nós habilitados com criptografia em trânsito usando o Openssl (Memcached)

Para acessar dados de quatro nós do ElastiCache Memcached habilitados com criptografia em trânsito, você precisa usar clientes que funcionem com Secure Socket Layer (SSL). Também é possível usar a Openssl `s_client` no Amazon Linux e no Amazon Linux 2.

Para usar a Openssl `s_client` para se conectar a um cluster Memcached habilitado com criptografia em trânsito no Amazon Linux 2 ou no Amazon Linux:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Criação de um cliente TLS Memcached usando Java

Para criar um cliente no modo TLS, faça o seguinte para inicializar o cliente com o apropriado: `SSLContext`

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

public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
        ConnectionFactoryBuilder();
```

```

    // Build SSLContext
    TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
    tmf.init((KeyStore) null);
    SSLContext sslContext = SSLContext.getInstance("TLS");
    sslContext.init(null, tmf.getTrustManagers(), null);
    // Create the client in TLS mode
    connectionFactoryBuilder.setSSLContext(sslContext);
    MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

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

```

Criação de um cliente TLS Memcached usando PHP

Para criar um cliente no modo TLS, faça o seguinte para inicializar o cliente com o apropriado: SSLContext

```

<?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/dg/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */

```

```
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastError_message(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSTextContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSTextContextConfig();
$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. */
$tls_client->createAndSetTLSTextContext((array)$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->getLastError_message(), "\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->getLastError_message(), "\n";
    exit(1);
}
```



```
}
```

Para obter mais informações sobre como usar o cliente PHP, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

Criptografia em repouso em ElastiCache

Para ajudar a manter seus dados seguros, a Amazon ElastiCache e o Amazon S3 oferecem maneiras diferentes de restringir o acesso aos dados em seu cache. Para obter mais informações, consulte [Amazon VPCs e ElastiCache segurança](#) e [Identity and Access Management para Amazon ElastiCache](#).

ElastiCache a criptografia em repouso é um recurso para aumentar a segurança dos dados criptografando dados em disco. Ela está sempre habilitada em um cache sem servidor. Quando habilitada, ela criptografa os seguintes aspectos:

- Disco durante as operações de sincronização, backup e swap
- Backups armazenados no Amazon S3

Os dados armazenados em SSDs (unidades de estado sólido) em clusters habilitados para armazenamento de dados em camadas são sempre criptografados.

ElastiCache oferece criptografia padrão (gerenciada pelo serviço) em repouso, bem como a capacidade de usar suas próprias chaves AWS KMS simétricas gerenciadas pelo cliente no [AWS Key Management Service \(KMS\)](#). Quando o backup do cache for feito, em opções de criptografia, escolha se você deseja usar a chave de criptografia padrão ou uma chave gerenciada pelo cliente. Para obter mais informações, consulte [Ativar criptografia em repouso](#).

Important

A habilitação da criptografia em repouso em um cluster do Valkey ou Redis OSS baseado em nós existente envolve a exclusão do grupo de replicação existente após a execução do backup e a restauração no grupo de replicação.

A criptografia em repouso só pode ser habilitada em um cache quando é criada. Como a criptografia e a descriptografia dos dados requerem processamento, a ativação da criptografia em repouso pode afetar o desempenho durante essas operações. Compare seus dados com e sem criptografia em repouso para determinar o impacto no desempenho para seus casos de uso.

Tópicos

- [Condições da criptografia em repouso](#)
- [Usando chaves gerenciadas pelo cliente do AWS KMS](#)

- [Ativar criptografia em repouso](#)
- [Consulte também](#)

Condições da criptografia em repouso

As seguintes restrições sobre a criptografia em ElastiCache repouso devem ser lembradas ao planejar sua implementação da ElastiCache criptografia em repouso:

- A criptografia em repouso é compatível em grupos de replicação que estejam executando as versões do Valkey 7.2 ou posteriores e do Redis OSS (3.2.6 programadas para EOL, consulte [Cronograma de fim de vida útil das versões do Redis OSS](#)) 4.0.10 ou posteriores.
- A criptografia em repouso tem suporte somente em grupos de replicação em execução em uma Amazon VPC.
- A criptografia em repouso é compatível somente com grupos de replicação que estejam executando os tipos de nó a seguir.
 - R7g, R6gd, R6g, R5, R4, R3
 - M7g, M6g, M5, M4, M3
 - T4g, T3, T2
 - C7gn

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- A criptografia em repouso é ativada por meio da configuração explícita do parâmetro `AtRestEncryptionEnabled` como `true`.
- Somente é possível habilitar a criptografia em repouso em um grupo de replicação ao criá-lo. Não é possível ativar e desativar a criptografia em repouso modificando um grupo de replicação. Para obter informações sobre a implementação de criptografia em repouso em um grupo de replicação, consulte [Ativar criptografia em repouso](#).
- Se um cluster estiver usando um tipo de nó da família r6gd, os dados armazenados em SSD serão criptografados independentemente de a criptografia em repouso estar ativada ou não.
- A opção de usar a chave gerenciada pelo cliente para criptografia em repouso não está disponível nas regiões AWS GovCloud (us-gov-east us-gov-west-1 e -1).
- Se um cluster estiver usando um tipo de nó da família r6gd, os dados armazenados no SSD serão criptografados com a chave AWS KMS gerenciada pelo cliente escolhida (ou criptografia gerenciada pelo serviço nas regiões).AWS GovCloud

- Com o Memcached, a criptografia em repouso só é compatível em caches de tecnologia sem servidor.
- Ao usar o Memcached, a opção de usar a chave gerenciada pelo cliente para criptografia em repouso não está disponível nas regiões AWS GovCloud (us-gov-east-1 e us-gov-west -1).

A implementação de criptografia em repouso pode reduzir o desempenho durante as operações de backup e sincronização de nós. Compare seus dados com criptografia em repouso e sem criptografia para determinar o impacto no desempenho da sua implementação.

Usando chaves gerenciadas pelo cliente do AWS KMS

ElastiCache suporta chaves AWS KMS simétricas gerenciadas pelo cliente (chave KMS) para criptografia em repouso. As chaves KMS gerenciadas pelo cliente são chaves de criptografia que você cria, possui e gerencia em sua conta.AWS Para obter mais informações, consulte [Chaves KMS da AWS](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS. As chaves devem ser criadas no AWS KMS antes de poderem ser usadas com ElastiCache.

Para saber como criar chaves raiz do AWS KMS, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

ElastiCache permite a integração com o AWS KMS. Para obter mais informações, consulte [Uso de concessões](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS. Nenhuma ação do cliente é necessária para permitir a ElastiCache integração da Amazon com o AWS KMS.

A chave de `kms:ViaService` condição limita o uso de uma chave AWS KMS (chave KMS) às solicitações de serviços especificados AWS. Para usar `kms:ViaService` com ElastiCache, inclua os dois `ViaService` nomes no valor da chave de condição: `elasticache.AWS_region.amazonaws.com` `dax.AWS_region.amazonaws.com` e. Para maiores informações, veja [kms: ViaService](#).

Você pode usar [AWS CloudTrail](#) para rastrear as solicitações que a Amazon ElastiCache envia AWS Key Management Service em seu nome. Todas as chamadas de API AWS Key Management Service relacionadas às chaves gerenciadas pelo cliente têm CloudTrail registros correspondentes. Você também pode ver as concessões ElastiCache criadas ao chamar a chamada da API [ListGrants](#)KMS.

Assim que um grupo de replicação é criptografado usando a chave gerenciada pelo cliente, todos os backups do grupo de replicação são criptografados da seguinte maneira:

- Os backups diários automáticos são criptografados usando a chave gerenciada pelo cliente associada ao cluster.

- O backup final criado quando o grupo de replicação é excluído também é criptografado usando a chave gerenciada pelo cliente associada ao grupo de replicação.
- Os backups criados manualmente são criptografados por padrão para usar a chave do KMS associada ao grupo de replicação. Você pode substituir escolhendo outra chave gerenciada pelo cliente.
- Por padrão, a cópia de um backup equivale a usar uma chave gerenciada pelo cliente associada ao backup de origem. Você pode substituir escolhendo outra chave gerenciada pelo cliente.

Note

- As chaves gerenciadas pelo cliente não podem ser usadas ao exportar backups para o bucket do Amazon S3 selecionado. No entanto, todos os backups exportados para o Amazon S3 são criptografados usando [Criptografia do lado do servidor](#). Você pode optar por copiar o arquivo de backup para um novo objeto do S3 e criptografar usando uma chave do KMS gerenciada pelo cliente, copiar o arquivo para outro bucket do S3 configurado com a criptografia padrão usando uma chave do KMS ou alterar uma opção de criptografia no próprio arquivo.
- Você também pode usar chaves gerenciadas pelo cliente para criptografar backups criados manualmente para grupos de replicação que não usem chaves gerenciadas pelo cliente para criptografia. Com essa opção, o arquivo de backup armazenado no Amazon S3 é criptografado usando uma chave do KMS, embora os dados não sejam criptografados no grupo de replicação original.

A restauração de um backup permite escolher entre as opções de criptografia disponíveis, semelhantes às opções de criptografia disponíveis ao criar um novo grupo de replicação.

- Se você excluir a chave ou [desabilitá-la](#) e [revogar as concessões](#) da chave que usou para criptografar um cache, o cache ficará irrecuperável. Em outras palavras, ele não pode ser modificado ou recuperado após uma falha de hardware. O AWS KMS exclui as chaves raiz somente após um período de espera de pelo menos sete dias. Depois que a chave for excluída, você poderá usar uma chave gerenciada pelo cliente diferente para criar um backup para fins de arquivamento.
- A rotação automática de chaves preserva as propriedades das chaves raiz do AWS KMS, portanto, a rotação não afeta sua capacidade de acessar seus ElastiCache dados. Os ElastiCache caches

criptografados da Amazon não oferecem suporte à rotação manual de chaves, o que envolve a criação de uma nova chave raiz e a atualização de qualquer referência à chave antiga. Para saber mais, consulte [Chaves rotativas do AWS KMS](#) no Guia do desenvolvedor do AWS Key Management Service.

- Criptografar um ElastiCache cache usando a chave KMS requer uma concessão por cache. Essa concessão é usada durante toda a vida útil do cache. Além disso, uma concessão por backup é usada durante a criação do backup. Essa concessão é retirada assim que o backup é criado.
- Para obter mais informações sobre concessões e limites do AWS KMS, consulte [Limites](#) no Guia do desenvolvedor do AWS Key Management Service.

Ativar criptografia em repouso

Todos os caches sem servidor têm criptografia em repouso habilitada.

Ao criar um cluster baseado em nós, você pode habilitar a criptografia em repouso definindo o parâmetro `AtRestEncryptionEnabled` como `true`. Não é possível ativar a criptografia em repouso em grupos de replicação existentes.

Você pode ativar a criptografia em repouso ao criar um ElastiCache cache. Você pode fazer isso usando a Console de gerenciamento da AWS, a AWS CLI, a ou a ElastiCache API.

Ao criar um cache, você pode escolher uma das seguintes opções:

- Default (Padrão) – Esta opção usa a criptografia gerenciada pelo serviço em repouso.
- Chave gerenciada pelo cliente — Essa opção permite que você forneça a chave ID/ARN do AWS KMS para criptografia em repouso.

Para saber como criar chaves raiz do AWS KMS, consulte [Criar chaves](#) no Guia do desenvolvedor do AWS Key Management Service

Sumário

- [Habilitando a criptografia em repouso usando o Console de gerenciamento da AWS](#)
- [Habilitando a criptografia em repouso usando o AWS CLI](#)

Habilitação da criptografia em repouso em um cluster do Valkey ou Redis OSS baseado em nós existente

Só é possível ativar a criptografia em repouso ao criar um grupo de replicação do Valkey ou Redis OSS. Se você tem um grupo de replicação no qual deseja ativar a criptografia em repouso, siga as etapas a seguir.

Para ativar a criptografia em repouso em um grupo de replicação existente

1. Crie um backup manual do seu grupo de replicação existente. Para obter mais informações, consulte [Realização de backups manuais](#).
2. Crie um novo grupo de replicação com base em um backup. No novo grupo de replicação, ative a criptografia em repouso. Para obter mais informações, consulte [Restauração de um backup para um novo cache](#).
3. Atualize os endpoints no seu aplicativo para apontarem para o novo grupo de replicação.
4. Exclua o grupo de replicação antigo. Para acessar mais informações, consulte [Excluindo um cluster no ElastiCache](#) ou [Exclusão de um grupo de replicação](#).

Habilitando a criptografia em repouso usando o Console de gerenciamento da AWS

Habilitação da criptografia em repouso em um cache sem servidor (console)

Todos os caches sem servidor têm criptografia em repouso habilitada. Por padrão, uma chave KMS AWS de propriedade própria é usada para criptografar dados. Para escolher sua própria AWS KMS chave, faça as seguintes seleções:

- Expanda a seção Visualizar configurações padrão.
- Escolha Personalizar configurações padrão na seção Visualizar configurações padrão.
- Escolha Personalize suas configurações de segurança na seção Segurança.
- Escolha CMK gerenciada pelo cliente na configuração Chave de criptografia.
- Selecione uma chave na configuração Chave AWS KMS.

Habilitação da criptografia em repouso em um cluster baseado em nós existente (console)

Ao projetar o próprio cache, as configurações “Dev/Teste” e “Produção” com o método “Criação fácil” têm a criptografia em repouso habilitada usando a chave Padrão. Ao escolher a configuração por conta própria, faça as seguintes seleções:

- Escolha a versão 3.2.6, 4.0.10 ou posterior como a versão do mecanismo.
- Clique na caixa de seleção ao lado de Habilitar para a opção Criptografia em repouso.
- Escolha uma Chave padrão ou uma CMK gerenciada pelo cliente.

Para o step-by-step procedimento, consulte o seguinte:

- [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#)
- [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)

Habilitando a criptografia em repouso usando o AWS CLI

Para habilitar a criptografia em repouso ao criar um cluster OSS Valkey ou Redis usando o AWS CLI, use o `at-rest-encryption-enabled` parâmetro `--` ao criar um grupo de replicação.

Habilitação da criptografia em repouso em um cluster do Valkey ou Redis OSS (modo cluster desabilitado) (CLI)

A operação a seguir cria o grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) `my-classic-rg` com três nós (`-- num-cache-clusters`), uma réplica primária e duas réplicas de leitura. A criptografia em repouso está habilitada para esse grupo de replicação (`-- at-rest-encryption-enabled`).

Os seguintes parâmetros e seus valores são necessários para ativar a criptografia neste grupo de replicação:

Principais parâmetros

- **`--engine`**: precisa ser `valkey` ou `redis`.
- **`--engine-version`**: se o mecanismo for Redis OSS, é preciso que seja 3.2.6, 4.0.10 ou posterior.
- **`--at-rest-encryption-enabled`**: obrigatório para habilitar a criptografia em repouso.

Example 1: cluster do Valkey ou Redis OSS (modo cluster desabilitado) com réplicas

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-classic-rg \  
  --engine valkey \  
  --engine-version 4.0.10 \  
  --num-cache-clusters 3 \  
  --at-rest-encryption-enabled
```



```
--replication-group-description "3 node replication group" \  
--cache-node-type cache.m4.large \  
--engine redis \  
--at-rest-encryption-enabled \  
--num-cache-clusters 3
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id my-classic-rg ^  
  --replication-group-description "3 node replication group" ^  
  --cache-node-type cache.m4.large ^  
  --engine redis ^  
  --at-rest-encryption-enabled ^  
  --num-cache-clusters 3 ^
```

Para obter informações adicionais, consulte:

- [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) do início \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitação da criptografia em repouso em um cluster do Valkey ou Redis OSS (modo cluster habilitado) (CLI)

A operação a seguir cria o grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) *my-clustered-rg* com três grupos de nós ou fragmentos (-- num-node-groups). Cada um tem três nós, uma réplica primária e duas réplicas de leitura (-- replicas-per-node-group). A criptografia em repouso está habilitada para esse grupo de replicação (-- at-rest-encryption-enabled).

Os seguintes parâmetros e seus valores são necessários para ativar a criptografia neste grupo de replicação:

Principais parâmetros

- **--engine**: precisa ser valkey ou redis.
- **--engine-version**: se o mecanismo for Redis OSS, é preciso que seja 4.0.10 ou posterior.
- **--at-rest-encryption-enabled**: obrigatório para habilitar a criptografia em repouso.

- **--cache-parameter-group:** deve ser `default-redis4.0.cluster.on` ou um derivado dele para torná-lo um grupo de replicação para o modo cluster habilitado.

Exemplo 2: um cluster do Valkey ou Redis OSS (modo cluster habilitado)

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-clustered-rg \  
  --replication-group-description "redis clustered cluster" \  
  --cache-node-type cache.m3.large \  
  --num-node-groups 3 \  
  --replicas-per-node-group 2 \  
  --engine redis \  
  --engine-version 6.2 \  
  --at-rest-encryption-enabled \  
  --cache-parameter-group default.redis6.x.cluster.on
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id my-clustered-rg ^  
  --replication-group-description "redis clustered cluster" ^  
  --cache-node-type cache.m3.large ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --engine redis ^  
  --engine-version 6.2 ^  
  --at-rest-encryption-enabled ^  
  --cache-parameter-group default.redis6.x.cluster.on
```

Para obter informações adicionais, consulte:

- [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster habilitado\) do início \(AWS CLI\)](#)
- [create-replication-group](#)

Consulte também

- [Amazon VPCs e ElastiCache segurança](#)

- [Identity and Access Management para Amazon ElastiCache](#)

Autenticação e autorização

AWS O Identity and Access Management (IAM) é um serviço da web que ajuda controlar o acesso aos recursos da AWS com segurança. O ElastiCache oferece suporte à autenticação de usuários usando o IAM e o comando AUTH do Valkey e Redis OSS, além de autorizar operações de usuários usando o controle de acesso baseado em função (RBAC).

Tópicos

- [Regras de controle de acesso com base em função \(RBAC\)](#)
- [Autenticar com o comando AUTH do Valkey e Redis OSS](#)
- [Desabilitar o controle de acesso em um cache do ElastiCache Valkey ou Redis OSS](#)

Regras de controle de acesso com base em função (RBAC)

Com o comando AUTH do Valkey e Redis OSS, conforme descrito em [Autenticar com o comando AUTH do Valkey e Redis OSS](#), você pode usar o controle de acesso baseado em funções (RBAC). O RBAC também é a única maneira de controlar o acesso a caches sem servidor. Isso está disponível para o Valkey 7.2 e posterior e o Redis OSS 6.0 a 7.2 e posterior.

O RBAC permite que você:

- Controle o acesso ao cache por meio de grupos de usuários. Esses grupos de usuários foram projetados como uma maneira de organizar o acesso a caches.
- Com o authN, tenha senhas por usuário em vez de tokens de autenticação por cluster.
- Com o authZ, tenha permissões de usuário refinadas.
- Baseie seu acesso ao cluster em ACLs.

Diferentemente do AUTH do Valkey e do Redis OSS, no qual todos os clientes autenticados têm acesso total ao cache em caso de autenticação do token, o RBAC permite atribuir usuários a sets dependendo das funções desejadas dos usuários. Esses sets foram projetados como uma maneira de organizar o acesso a caches.

Com o RBAC, você cria usuários e atribui a eles permissões específicas usando uma cadeia de acesso, conforme descrito a seguir. Você atribui os usuários a conjuntos alinhados com uma função específica (administradores, recursos humanos) que são então implantados em um ou mais caches.

ElastiCache Fazendo isso, você pode estabelecer limites de segurança entre clientes usando o mesmo cache ou caches do Valkey ou Redis OSS e impedir que os clientes acessem os dados uns dos outros.

O RBAC foi projetado para oferecer suporte à introdução da [ACL](#) no Redis OSS 6. Quando você usa o RBAC com seu cache OSS ElastiCache Valkey ou Redis, há algumas limitações:

- Um grupo de usuários configurado para o mecanismo “VALKEY” só pode conter usuários que estejam usando um mecanismo de autenticação (senha ou IAM). Isso significa que todos os usuários com o mecanismo “VALKEY” e quaisquer outros usuários com o mecanismo “Redis” que tenham sua configuração definida para autenticação com senha ou IAM podem estar nesse grupo de usuários.
- Ao usar o RBAC com clusters Valkey, os dois grupos de usuários com o mecanismo “VALKEY” e com o mecanismo “REDIS” podem ser usados.
- Ao usar o RBAC com clusters Redis OSS, você poderá usar somente grupos de usuários com o mecanismo “REDIS”.
- Não é possível especificar senhas em uma string de acesso. Você define senhas com [CreateUser](#) e [ModifyUser](#) chamadas.
- Para direitos de usuário, você habilita ou desabilita usuários com `on` e `off` como parte da string de acesso. Se nenhum deles for especificado na string de acesso, o usuário receberá `off` e não terá direitos de acesso ao cache.
- Não é possível usar comandos proibidos e renomeados como parte da string de acesso. Se você especificar um comando proibido ou renomeado, será emitida uma exceção. Se você quiser usar listas de controle de acesso (ACLs) para um comando renomeado, especifique o nome original do comando, em outras palavras, o nome do comando antes de ser renomeado.
- Não é possível usar o comando `reset` como parte de uma string de acesso. Você especifica senhas com parâmetros de API e, ElastiCache para Valkey e Redis, o OSS gerencia senhas. Assim, você não pode usar `reset` porque ele removeria todas as senhas de um usuário.
- O Redis OSS 6 apresenta o comando [ACL LIST](#). Esse comando retorna uma lista de usuários junto com as regras de ACL aplicadas a cada usuário. ElastiCache suporta o `ACL LIST` comando, mas não inclui suporte para hashes de senha, como faz o Redis OSS. Com ElastiCache, você pode usar a [DescribeUsers](#) operação para obter informações semelhantes, incluindo as regras contidas na string de acesso. No entanto, [DescribeUsers](#) não recupera a senha do usuário.
- [Outros comandos somente ElastiCache para leitura compatíveis com o Valkey e o Redis OSS incluem ACL WHOAMI, ACL USERS e ACL CAT.](#) ElastiCache para Valkey e Redis, o OSS não oferece suporte a nenhum outro comando ACL baseado em gravação.

- Os limites a seguir se aplicam ao seguinte:

Recurso	Máximo permitido
Usuários por grupo de usuários	100
Número de DPU's	1000
Número de grupos de usuários	100

RBAC com Valkey

Ao usar o Controle de Acesso Baseado em Função com o Valkey, os usuários e grupos de usuários são criados com o tipo de mecanismo “VALKEY”. Isso é recomendado, pois, por padrão, o Valkey com RBAC oferece maior segurança em comparação com o Redis OSS. Os clusters Valkey provisionados e com tecnologia sem servidor oferecem suporte a associações de usuários e grupos de usuários do VALKEY.

Os principais atributos do Valkey Access Control incluem:

- Os usuários do Valkey estão restritos somente às associações de grupos de usuários do Valkey.
- Os grupos de usuários do Valkey podem conter usuários do Valkey e usuários do Redis OSS protegidos por senha ou habilitados para autenticação do IAM.
- Os usuários do Valkey devem usar proteção por senha ou autenticação do IAM.
- Os grupos de usuários do VALKEY só podem ser associados aos clusters do VALKEY
- Não há nenhum requisito de usuário padrão. Quando o grupo de usuários do Valkey é anexado aos clusters, o requisito de usuário padrão é automaticamente desativado. Os clientes verão que o usuário padrão está desativado ao usar o comando ACL LIST.

Veja a seguir mais informações sobre como usar o RBAC com o ElastiCache Valkey e o Redis OSS.

Tópicos

- [Especificação de permissões usando uma string de acesso](#)
- [Aplicando o RBAC a um cache ElastiCache para Valkey ou Redis OSS](#)
- [Migração do AUTH para o RBAC](#)
- [Migração do RBAC para o AUTH](#)

- [Alternância automática de senhas para os usuários](#)
- [Autenticação com o IAM](#)

Especificação de permissões usando uma string de acesso

Para especificar permissões para um cache OSS ElastiCache Valkey ou Redis, você cria uma string de acesso e a atribui a um usuário por meio do ou.AWS CLIConsole de gerenciamento da AWS

As cadeias de caracteres de acesso são definidas como uma lista de regras delimitadas por espaço que são aplicadas ao usuário. Eles definem quais comandos um usuário pode executar e em quais chaves um usuário pode operar. Para executar um comando, um usuário deve ter acesso ao comando que está sendo executado e todas as chaves que estão sendo acessadas pelo comando. Regras são aplicadas da esquerda para a direita cumulativamente, e uma string mais simples pode ser usada em vez da fornecida se houver redundâncias na string fornecida.

Para obter informações sobre a sintaxe das regras ACL, consulte [ACL](#).

No exemplo a seguir, a string de acesso representa um usuário ativo com acesso a todas as chaves e comandos disponíveis.

```
on ~* +@all
```

A sintaxe da cadeia de acesso é dividida da seguinte forma:

- on: o usuário é um usuário ativo.
- ~*: o acesso é dado a todas as chaves disponíveis.
- +@all: o acesso é dado a todos os comandos disponíveis.

As configurações anteriores são as menos restritivas. Você pode modificar essas configurações para torná-las mais seguras.

No exemplo a seguir, a string de acesso representa um usuário com acesso restrito ao acesso de leitura em chaves que começam com o keypace "app::"

```
on ~app::* -@all +@read
```

Você pode refinar mais essas permissões listando comandos aos quais o usuário tem acesso:

+*command1*: o acesso do usuário aos comandos é limitado a *command1*.

+@category: o acesso do usuário é limitado a uma categoria de comandos.

Para obter informações sobre como atribuir uma string de acesso a um usuário, consulte [Criação de usuários e grupos de usuários com o console e a CLI](#).

Se você estiver migrando uma carga de trabalho existente para ElastiCache, poderá recuperar a string de acesso chamando `ACL LIST`, excluindo o usuário e quaisquer hashes de senha.

Para o Redis OSS versão 6.2 e posteriores, a seguinte sintaxe de string de acesso também tem suporte:

- `&*`: o acesso é dado a todos os canais disponíveis.

Para o Redis OSS versão 7.0 e posteriores, a seguinte sintaxe de string de acesso também tem suporte:

- `|`: pode ser usado para bloquear subcomandos (por exemplo, `"-config|set"`).
- `%R~<pattern>`: adicione o padrão de chave de leitura especificado. Comporta-se de forma semelhante ao padrão de chave normal, mas só concede permissão para ler chaves que correspondam ao padrão fornecido. Para obter mais informações, consulte [permissões de chave](#).
- `%W~<pattern>`: adicione o padrão de chave de gravação especificado. Comporta-se de forma semelhante ao padrão de chave normal, mas só concede permissão para gravar em chaves que correspondam ao padrão fornecido. Para obter mais informações, consulte [Permissões de chave de ACL](#).
- `%RW~<pattern>`: alias para `~<pattern>`.
- `(<rule list>)`: crie um novo seletor para combinar as regras. Os seletores são avaliados após as permissões do usuário e são avaliados de acordo com a ordem em que são definidos. Se um comando corresponder às permissões do usuário ou a qualquer seletor, ele será permitido. Para obter mais informações, consulte [Seletores de ACL](#).
- `clearselectors`: exclua todos os seletores associados ao usuário.

Aplicando o RBAC a um cache ElastiCache para Valkey ou Redis OSS

ElastiCache Para usar o Valkey ou o Redis OSS RBAC, siga as seguintes etapas:

1. Crie um ou mais usuários.
2. Crie um grupo de usuários e adicione usuários ao grupo.
3. Atribua o grupo de usuários a um cache que tenha criptografia em trânsito habilitada.

Essas etapas estão descritas em detalhes a seguir.

Tópicos

- [Criação de usuários e grupos de usuários com o console e a CLI](#)
- [Gerenciamento de grupos de usuários com o console e a CLI](#)
- [Atribuição de grupos de usuários a caches sem servidor](#)
- [Atribuição de grupos de usuários a grupos de replicação](#)

Criação de usuários e grupos de usuários com o console e a CLI

As informações de usuário para usuários do RBAC são um ID de usuário, nome de usuário e, opcionalmente, uma senha e uma string de acesso. A string de acesso fornece o nível de permissão em chaves e comandos. O ID de usuário é exclusivo para o usuário e o nome de usuário é o que é passado para o mecanismo.

Certifique-se de que as permissões do usuário fornecidas fazem sentido com a finalidade pretendida do grupo de utilizadores. Por exemplo, se você criar um grupo de usuários chamado `Administrators`, qualquer usuário que você adicionar a esse grupo deve ter sua string de acesso definida para acesso total a chaves e comandos. Para usuários em um grupo de usuários e-commerce, você pode definir suas cadeias de acesso para acesso somente leitura.

ElastiCache configura automaticamente um usuário padrão com ID de usuário e nome de usuário `default` e o adiciona a todos os grupos de usuários. Você não pode excluir ou modificar esse usuário. Esse usuário destina-se à compatibilidade com o comportamento padrão das versões anteriores do Redis OSS e tem uma string de acesso que permite chamar todos os comandos e acessar todas as chaves.

Para adicionar um controle de acesso indicado a um cache, substitua esse usuário padrão por um novo que não esteja habilitado ou use uma senha forte. Para alterar o usuário padrão, crie um novo usuário com o nome de usuário definido como `default`. Em seguida, você pode trocá-lo pelo usuário padrão original.

Os seguintes procedimentos mostram como trocar o usuário `default` original por outro usuário `default` que tem uma string de acesso modificada.

Para modificar o usuário padrão no console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.

2. Escolha Gerenciamento de grupos de usuários no painel de navegação.
3. Em ID do grupo de usuários, escolha o ID que você deseja modificar. Verifique se você escolheu o link e não a caixa de seleção.
4. Escolha Modificar.
5. Na janela Modificar, escolha Gerenciar. Para “selecionar o usuário que você deseja”, selecione o usuário com o nome de usuário como padrão.
6. Selecione Choose (Escolher).
7. Escolha Modificar. Quando você fizer isso, todas as conexões existentes com um cache que o usuário padrão original tiver serão encerradas.

Para modificar o usuário padrão com o AWS CLI

1. Criar um novo usuário com o nome de usuário default usando os seguintes comandos.

Para Linux, macOS ou Unix:

```
aws elasticache create-user \  
  --user-id "new-default-user" \  
  --user-name "default" \  
  --engine "VALKEY" \  
  --passwords "a-strong-password" \  
  --access-string "off +get ~keys*"
```

Para Windows:

```
aws elasticache create-user ^  
  --user-id "new-default-user" ^  
  --user-name "default" ^  
  --engine "VALKEY" ^  
  --passwords "a-strong-password" ^  
  --access-string "off +get ~keys*"
```

2. Crie um grupo de usuários e adicione o usuário que você criou anteriormente.

Para Linux, macOS ou Unix:

```
aws elasticache create-user-group \  
  --user-group-id "new-group-2" \  
  --engine "VALKEY" \  
  --passwords "a-strong-password" \  
  --access-string "off +get ~keys*"
```

```
--user-ids "new-default-user"
```

Para Windows:

```
aws elasticache create-user-group ^  
  --user-group-id "new-group-2" ^  
  --engine "VALKEY" ^  
  --user-ids "new-default-user"
```

Ao criar um usuário, você pode configurar até duas senhas. Quando você modifica uma senha, todas as conexões existentes com os caches são mantidas.

Em particular, esteja ciente dessas restrições de senha de usuário ao usar o RBAC ElastiCache para Valkey e Redis OSS:

- As senhas devem ter de 16 a 128 caracteres imprimíveis.
- Os seguintes caracteres não alfanuméricos não são permitidos: , " " / @.

Gerenciamento de usuários com o console e a CLI

Use o procedimento a seguir para gerenciar usuários no console.

Para gerenciar usuários no console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No ElastiCache painel da Amazon, escolha Gerenciamento de usuários. As seguintes opções estão disponíveis:
 - Criar usuário: ao criar um usuário, você insere um ID de usuário, um nome de usuário, um modo de autenticação e uma string de acesso. A string de acesso define o nível de permissão para quais chaves e comandos o usuário é permitido.

Ao criar um usuário, você pode configurar até duas senhas. Quando você modifica uma senha, todas as conexões existentes com os caches são mantidas.

- Modificar usuário: permite a você atualizar as configurações de um usuário ou alterar a string de acesso.

- Excluir usuário: a conta de usuário será removida de todos os grupos de usuários aos quais ela pertence.

Use o procedimento a seguir para gerenciar usuários na AWS CLI.

Para modificar um usuário usando a CLI

- Use o comando `modify-user` para atualizar a senha ou senhas de um usuário ou alterar as permissões de acesso de um usuário.

Quando um usuário é modificado, os grupos de usuários associados ao usuário são atualizados com todos os caches associados ao grupo de usuários. Todas as conexões existentes são mantidas. Veja os exemplos a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-user \  
  --user-id user-id-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --authentication-mode Type=iam
```

Para Windows:

```
aws elasticache modify-user ^  
  --user-id user-id-1 ^  
  --access-string "~objects:* ~items:* ~public:*" ^  
  --authentication-mode Type=iam
```

Note

Não recomendamos usar a opção `nopass`. Se o fizer, recomendamos definir as permissões do usuário como somente de leitura, com acesso a um conjunto limitado de chaves.

Para excluir um usuário usando a CLI

- Use o comando `delete-user` para excluir um usuário. A conta é excluída e removida de todos os grupos de usuários aos quais pertence. Veja um exemplo do a seguir:

Para Linux, macOS ou Unix:

```
aws elasticache delete-user \  
--user-id user-id-2
```

Para Windows:

```
aws elasticache delete-user ^  
--user-id user-id-2
```

Para ver uma lista de usuários, execute a operação [describe-users](#).

```
aws elasticache describe-users
```

Gerenciamento de grupos de usuários com o console e a CLI

É possível criar grupos de usuários para organizar e controlar o acesso de usuários a um ou mais caches, conforme mostrado a seguir.

Use o procedimento a seguir para gerenciar grupos de usuários no console.

Como gerenciar grupos de usuários usando o console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No ElastiCache painel da Amazon, escolha Gerenciamento de grupos de usuários.

As operações a seguir estão disponíveis para criar novos grupos de usuários:

- Criar: ao criar um grupo de usuários, você adiciona usuários e, em seguida, atribui os grupos de usuários a caches. Por exemplo, é possível criar um grupo de usuários Admin para usuários que tenham funções administrativas em um cache.

 Important

Se você não estiver usando um grupo de usuários Valkey ou Redis OSS, inclua um usuário padrão ao criar um grupo de usuários.

- **Add Users (Adicionar usuários):** adiciona usuários ao grupo de usuários.
- **Remove Users (Remover usuários):** remove usuários do grupo de usuários. Quando os usuários são removidos de um grupo de usuários, todas as conexões existentes que eles têm com um cache são encerradas.
- **Delete (Excluir):** use para excluir um grupo de usuários. Observe que o próprio grupo de usuários, e não os usuários pertencentes ao grupo, serão excluídos.

Para grupos de usuários existentes, você pode fazer o seguinte:

- **Add Users (Adicionar usuários):** adiciona usuários existentes ao grupo de usuários.
- **Delete Users (Excluir usuários):** remove usuários existentes do grupo de usuários.

 Note

Os usuários são removidos do grupo de usuários, mas não excluídos do sistema.

Use os procedimentos a seguir para gerenciar grupos de usuários na CLI.

Para criar um novo grupo de usuários e adicionar um usuário usando a CLI

- Use o comando `create-user-group`, conforme mostrado a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache create-user-group \  
  --user-group-id "new-group-1" \  
  --engine "VALKEY" \  
  --user-ids user-id-1, user-id-2
```

Para Windows:

```
aws elasticache create-user-group ^  
  --user-group-id "new-group-1" ^  
  --engine "VALKEY" ^  
  --user-ids user-id-1, user-id-2
```

Para modificar um grupo de usuários adicionando novos usuários ou removendo membros atuais usando a CLI

- Use o comando `modify-user-group`, conforme mostrado a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-user-group --user-group-id new-group-1 \  
--user-ids-to-add user-id-3 \  
--user-ids-to-remove user-id-2
```

Para Windows:

```
aws elasticache modify-user-group --user-group-id new-group-1 ^  
--user-ids-to-add user-id-3 ^  
--user-ids-to-remove user-id-2
```

Note

Quaisquer conexões abertas pertencentes a um usuário removido de um grupo de usuários são encerradas por este comando.

Para excluir um grupo de usuários usando a CLI

- Use o comando `delete-user-group`, conforme mostrado a seguir. O grupo de usuários em si, não os usuários pertencentes ao grupo, é excluído.

Para Linux, macOS ou Unix:

```
aws elasticache delete-user-group /  
--user-group-id
```

Para Windows:

```
aws elasticache delete-user-group ^  
--user-group-id
```

Para ver uma lista de grupos de usuários, você pode chamar a [describe-user-groups](#) operação.

```
aws elasticache describe-user-groups \  
  --user-group-id test-group
```

Atribuição de grupos de usuários a caches sem servidor

Depois que você tiver criado um grupo de usuários e adicionado usuários, a etapa final na implementação do RBAC será atribuir o grupo de usuários a um cache sem servidor.

Atribuição de grupos de usuários a caches sem servidor usando o console

Para adicionar um grupo de usuários a um cache sem servidor usando o Console de gerenciamento da AWS, faça o seguinte:

- Para o modo cluster desabilitado, consulte [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#)
- Para o modo cluster habilitado, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)

Atribuindo grupos de usuários a caches sem servidor usando o AWS CLI

A AWS CLI operação a seguir cria um cache sem servidor usando o `user-group-id` parâmetro com o valor `my-user-group-id`. Substitua o grupo de sub-redes `sng-test` por um grupo de sub-redes que exista.

Principais parâmetros

- **--engine**: precisa ser VALKEY ou REDIS.
- **--user-group-id**: este valor fornece o ID do grupo de usuários, composto de usuários com permissões de acesso especificadas para o cache.

Para Linux, macOS ou Unix:

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name "new-serverless-cache" \  
  --description "new-serverless-cache" \  
  --engine "VALKEY" \  
  --user-group-id my-user-group-id
```

```
--user-group-id "new-group-1"
```

Para Windows:

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name "new-serverless-cache" ^  
  --description "new-serverless-cache" ^  
  --engine "VALKEY" ^  
  --user-group-id "new-group-1"
```

A AWS CLI operação a seguir modifica um cache sem servidor com o user-group-id parâmetro com o valor. *my-user-group-id*

Para Linux, macOS ou Unix:

```
aws elasticache modify-serverless-cache \  
  --serverless-cache-name serverless-cache-1 \  
  --user-group-id "new-group-2"
```

Para Windows:

```
aws elasticache modify-serverless-cache ^  
  --serverless-cache-name serverless-cache-1 ^  
  --user-group-id "new-group-2"
```

Eventuais modificações feitas em um cache são atualizadas de maneira assíncrona. Você pode monitorar este progresso visualizando os eventos. Para obter mais informações, consulte [Visualizando ElastiCache eventos](#).

Atribuição de grupos de usuários a grupos de replicação

Depois de criar um grupo de usuários e adicionar usuários, a etapa final na implementação do RBAC é atribuir o grupo de usuários a um grupo de replicação.

Atribuição de grupos de usuários a grupos de replicação usando o console

Para adicionar um grupo de usuários a uma replicação usando o Console de gerenciamento da AWS, faça o seguinte:

- Para o modo cluster desabilitado, consulte [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#)

- Para o modo cluster habilitado, consulte [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#)

Atribuindo grupos de usuários a grupos de replicação usando o AWS CLI

A AWS CLI operação a seguir cria um grupo de replicação com a criptografia em trânsito (TLS) ativada e o `user-group-ids` parâmetro com o valor `my-user-group-id`. Substitua o grupo de sub-redes `sng-test` por um grupo de sub-redes que exista.

Principais parâmetros

- **--engine**: precisa ser `valkey` ou `redis`.
- **--engine-version**: deve ser 6.0 ou posterior.
- **--transit-encryption-enabled**: necessário para autenticação e para associar um grupo de usuários.
- **--user-group-ids**: este valor fornece o ID do grupo de usuários, composto de usuários com permissões de acesso especificadas para o cache.
- **--cache-subnet-group**: necessário para associar um grupo de usuários.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id "new-replication-group" \  
  --replication-group-description "new-replication-group" \  
  --engine "VALKEY" \  
  --cache-node-type cache.m5.large \  
  --transit-encryption-enabled \  
  --user-group-ids "new-group-1" \  
  --cache-subnet-group "cache-subnet-group"
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id "new-replication-group" ^  
  --replication-group-description "new-replication-group" ^  
  --engine "VALKEY" ^  
  --cache-node-type cache.m5.large ^  
  --transit-encryption-enabled ^  
  --user-group-ids "new-group-1" ^
```

```
--cache-subnet-group "cache-subnet-group"
```

A AWS CLI operação a seguir modifica um grupo de replicação com a criptografia em trânsito (TLS) ativada e o `user-group-ids` parâmetro com o valor `my-user-group-id`

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id replication-group-1 \  
  --user-group-ids-to-remove "new-group-1" \  
  --user-group-ids-to-add "new-group-2"
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id replication-group-1 ^  
  --user-group-ids-to-remove "new-group-1" ^  
  --user-group-ids-to-add "new-group-2"
```

Observe o `PendingChanges` na resposta. Eventuais modificações feitas em um cache são atualizadas de maneira assíncrona. Você pode monitorar este progresso visualizando os eventos. Para obter mais informações, consulte [Visualizando ElastiCache eventos](#).

Migração do AUTH para o RBAC

Se você estiver usando o AUTH conforme descrito em [Autenticar com o comando AUTH do Valkey e Redis OSS](#) e deseja migrar para usar o RBAC, use os seguintes procedimentos.

Use o procedimento a seguir para migrar do AUTH para o RBAC usando o console.

Para migrar do AUTH do Valkey ou Redis OSS para o RBAC usando o console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região em que o cache que você deseja modificar está localizado.
3. No painel de navegação, escolha o mecanismo em execução no cache que você deseja modificar.

Uma lista dos caches do mecanismo escolhido é exibida.

4. Na lista de caches, para o cache que você deseja modificar, escolha o nome.
5. Para Actions (Ações), escolha Modify (Modificar).

A janela Modificar é exibida.

6. Para Controle de acesso, escolha Lista de controle de acesso do grupo de usuários.
7. Em Lista de controle de acesso do grupo de usuários, escolha um grupo de usuários.
8. Escolha Previsualizar alterações e, na próxima tela, Modificar.

Use o procedimento a seguir para migrar do AUTH do Valkey ou Redis OSS para o RBAC usando a CLI.

Para migrar do AUTH para o RBAC usando a CLI

- Use o comando `modify-replication-group`, conforme mostrado a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group --replication-group-id test \  
--auth-token-update-strategy DELETE \  
--user-group-ids-to-add user-group-1
```

Para Windows:

```
aws elasticache modify-replication-group --replication-group-id test ^  
--auth-token-update-strategy DELETE ^  
--user-group-ids-to-add user-group-1
```

Migração do RBAC para o AUTH

Se você estiver usando o RBAC e desejar migrar para o AUTH do Redis OSS, consulte [Migração do RBAC para o AUTH](#).

Note

Se precisar desativar o controle de acesso em um ElastiCache cache, você precisará fazer isso por meio do AWS CLI. Para obter mais informações, consulte [the section called “Desabilitar o controle de acesso em um cache do ElastiCache Valkey ou Redis OSS”](#).

Alternância automática de senhas para os usuários

Com AWS Secrets Manager, você pode substituir as credenciais codificadas em seu código (incluindo senhas) por uma chamada de API para o Secrets Manager para recuperar o segredo programaticamente. Isso ajuda a garantir que o segredo não será comprometido por alguém que esteja examinando seu código, pois o segredo simplesmente não está ali. Além disso, configure o Secrets Manager para alterar automaticamente o segredo para você de acordo com a programação que você especificar. Isso permite substituir segredos de longo prazo por outros de curto prazo, ajudando a reduzir de maneira significativa o risco de comprometimento.

Usando o Secrets Manager, você pode alternar automaticamente suas ElastiCache senhas do OSS do Redis (ou seja, segredos) usando uma AWS Lambda função fornecida pelo Secrets Manager.

Para obter mais informações sobre AWS Secrets Manager, consulte [O que é AWS Secrets Manager?](#)

Como ElastiCache usa segredos

O Valkey 7.2 e superiores têm um conjunto de atributos equivalente ao Redis OSS 7.0. No Redis OSS 6, ElastiCache introduziu [Regras de controle de acesso com base em função \(RBAC\)](#) para proteger o cluster Valkey ou Redis OSS. Esse recurso permite que certas conexões sejam limitadas em termos dos comandos que podem ser executados e das chaves que podem ser acessadas. Com o RBAC, enquanto o cliente cria um usuário com senhas, os valores da senha precisam ser inseridos manualmente em texto sem formatação e ficam visíveis para o operador.

Com o Secrets Manager, as aplicações buscam a senha do Secrets Manager em vez de inseri-la manualmente e armazená-la na configuração da aplicação. Para obter informações sobre como fazer isso, consulte [Como ElastiCache os usuários são associados ao segredo](#).

O uso de segredos tem um custo. Para obter informações sobre preços, consulte [Preços do AWS](#).

Como ElastiCache os usuários são associados ao segredo

O Secrets Manager manterá uma referência para o usuário associado no campo `SecretString` do segredo. Não haverá nenhuma referência ao segredo do ElastiCache lado de fora.

```
{
  "password": "strongpassword",
  "username": "user1",
  "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1" //this is the
  bond between the secret and the user
}
```

Função de alternância do Lambda

Para habilitar a alternância automática de senhas do Secrets Manager, você criará uma função do Lambda que interagirá com a API [modify-user](#) para atualizar as senhas do usuário.

Para obter informações sobre como isso funciona, consulte [Como funciona a alternância](#).

Note

Para alguns AWS serviços, para evitar o cenário confuso de substituto, AWS recomenda que você use as chaves de condição `aws:SourceArn` e as chaves de condição `aws:SourceAccount` globais. No entanto, se você incluir a condição `aws:SourceArn` em sua política de função de alternância, a função de alternância só poderá ser usada para alternar o segredo especificado por esse ARN. Recomendamos que inclua apenas a chave de contexto `aws:SourceAccount`, de modo que possa usar a função de alternância para vários segredos.

Para quaisquer problemas que você possa encontrar, consulte [Solucionar problemas de rotação do AWS Secrets Manager](#).

Como criar um ElastiCache usuário e associá-lo ao Secrets Manager

As seguintes etapas ilustram como criar um usuário e associá-lo ao Secrets Manager:

1. Criar um usuário inativo

Para Linux, macOS ou Unix:

```
aws elasticache create-user \  
  --user-id user1 \  
  --user-name user1 \  
  --engine "REDIS" \  
  --no-password \ // no authentication is required  
  --access-string "*off* +get ~keys*" // this disables the user
```

Para Windows:

```
aws elasticache create-user ^  
  --user-id user1 ^  
  --user-name user1 ^
```

```
--engine "REDIS" ^  
--no-password ^ // no authentication is required  
--access-string "*off* +get ~keys*" // this disables the user
```

Você verá uma resposta semelhante ao seguinte:

```
{  
  "UserId": "user1",  
  "UserName": "user1",  
  "Status": "active",  
  "Engine": "redis",  
  "AccessString": "off ~keys* -@all +get",  
  "UserGroupIds": [],  
  "Authentication": {  
    "Type": "no_password"  
  },  
  "ARN": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"  
}
```

2. Criar um segredo

Para Linux, macOS ou Unix:

```
aws secretsmanager create-secret \  
--name production/ec/user1 \  
--secret-string \  
'{  
  "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",  
  "username": "user1"  
}'
```

Para Windows:

```
aws secretsmanager create-secret ^  
--name production/ec/user1 ^  
--secret-string ^  
'{  
  "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",  
  "username": "user1"  
}'
```

Você verá uma resposta semelhante ao seguinte:

```
{
  "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
  "Name": "production/ec/user1",
  "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

3. Configurar uma função do Lambda para alternar sua senha

- a. Faça login no Console de gerenciamento da AWS e abra o console Lambda em <https://console.aws.amazon.com/lambda/>
- b. No painel de navegação, escolha Functions (Funções) e escolha a função que você criou. Escolha o nome da função, não a caixa de seleção à esquerda.
- c. Escolha a guia Configuration (Configuração).
- d. Em General configuration (Configuração geral), escolha Edit (Editar) e defina Timeout (Tempo limite) para pelo menos 12 minutos.
- e. Escolha Salvar.
- f. Escolha Environment variables (Variáveis de ambiente) e defina o seguinte:
 - i. SECRETS_MANAGER_ENDPOINT – <https://secretsmanager.REGION.amazonaws.com>
 - ii. SECRET_ARN – O nome de recurso da Amazon (ARN) do segredo que você criou na etapa 2.
 - iii. USER_NAME — Nome de usuário do ElastiCache usuário,
 - iv. Escolha Salvar.
- g. Escolha Permissions (Permissões)
- h. Em Execution role (Função de execução), escolha o nome da função do Lambda para exibição no console do IAM.
- i. A função do Lambda precisará da seguinte permissão para modificar os usuários e definir a senha:

ElastiCache

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:DescribeUsers",
        "elasticache:ModifyUser"
      ],
      "Resource": "arn:aws:elasticache:us-east-1:123456789012:user:user1"
    }
  ]
}
```

Secrets Manager

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:UpdateSecretVersionStage"
      ],
      "Resource": "arn:aws:secretsmanager:us-east-1:123456789012:secret:XXXX"
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetRandomPassword",
      "Resource": "*"
    }
  ]
}
```



```
}  
]  
}
```

4. Configurar a alternância de segredos do Secrets Manager

- a. Usando o Console de gerenciamento da AWS, consulte [Configurar a rotação automática para AWS segredos do Secrets Manager usando o console](#)

Para obter mais informações sobre como configurar uma programação de alternância, consulte [Programar expressões nas alternâncias do Secrets Manager](#).

- b. Usando o AWS CLI, consulte [Configurar a rotação automática para AWS Secrets Manager usar o AWS Command Line Interface](#)

Autenticação com o IAM

Tópicos

- [Visão geral do](#)
- [Limitações](#)
- [Configuração](#)
- [Conexão](#)

Visão geral do

Com a autenticação IAM, você pode autenticar uma conexão ElastiCache para Valkey ou Redis OSS usando identidades AWS IAM, quando seu cache está configurado para usar Valkey ou Redis OSS versão 7 ou superior. Isso possibilita que você fortaleça seu modelo de segurança e simplifique várias tarefas administrativas de segurança. Você também pode usar a autenticação do IAM para configurar um controle de acesso refinado para cada ElastiCache cache e ElastiCache usuário individual, seguindo os princípios de permissões de privilégio mínimo. O IAM Authentication for ElastiCache funciona fornecendo um token de autenticação IAM de curta duração em vez de uma senha de ElastiCache usuário de longa duração no OSS ou comando Valkey ou Redis. AUTH HELLO Para obter mais informações sobre o token de autenticação do IAM, consulte o [processo de assinatura do Signature versão 4](#) no Guia de referência AWS geral e no exemplo de código abaixo.

Você pode usar as identidades do IAM e suas políticas associadas para restringir ainda mais o acesso ao Valkey ou Redis OSS. Também é possível conceder acesso a usuários dos provedores de identidade federados diretamente para caches do Valkey ou Redis OSS.

Para usar o AWS IAM com ElastiCache, primeiro você precisa criar um ElastiCache usuário com o modo de autenticação definido como IAM. Em seguida, você pode criar ou reutilizar uma identidade do IAM. A identidade do IAM precisa de uma política associada para conceder a `elasticache:Connect` ação ao ElastiCache cache e ao ElastiCache usuário. Depois de configurado, você pode criar um token de autenticação do IAM usando AWS as credenciais do usuário ou da função do IAM. Por fim, você precisa fornecer o token de autenticação do IAM de curta duração como uma senha no cliente do Valkey ou Redis OSS ao se conectar ao cache. Um cliente Valkey ou Redis OSS com suporte para provedor de credenciais pode gerar automaticamente as credenciais temporárias automaticamente para cada nova conexão. ElastiCache realizará a autenticação do IAM para solicitações de conexão de ElastiCache usuários habilitados para IAM e validará as solicitações de conexão com o IAM.

Limitações

Ao usar a autenticação do IAM, as seguintes limitações se aplicam:

- A autenticação do IAM está disponível quando usada ElastiCache para Valkey 7.2 e superior ou Redis OSS versão 7.0 e superior.
- Para ElastiCache usuários habilitados para IAM, as propriedades de nome de usuário e ID de usuário devem ser idênticas.
- O token de autenticação do IAM é válido por 15 minutos. Para conexões de longa duração, recomendamos usar um cliente do Valkey ou Redis OSS que suporte uma interface de provedor de credenciais.
- Uma conexão autenticada pelo IAM ElastiCache para Valkey ou Redis OSS será automaticamente desconectada após 12 horas. A conexão pode ser prolongada por 12 horas enviando um comando AUTH ou HELLO com um novo token de autenticação do IAM.
- A autenticação do IAM não tem suporte em comandos MULTI EXEC.
- Atualmente, a autenticação do IAM não dá suporte às seguintes chaves de contexto de condição global:
 - Durante o uso da autenticação do IAM com caches sem servidor, `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` e `aws:ResourceTag/%s` (de caches sem servidor associados e usuários) são compatíveis.
 - Durante o uso da autenticação do IAM com grupos de replicação, `aws:SourceIp` e `aws:ResourceTag/%s` (de grupos de replicação associados e usuários) são compatíveis.

Para obter mais informações sobre chaves de contexto de condição global, consulte [Chaves de contexto de condição global da AWS](#) no Guia do usuário do IAM.

Configuração

Como configurar a autenticação do IAM

1. Criar um cache

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

2. Crie um documento de política de confiança do IAM, conforme mostrado abaixo, para o perfil que permita que sua conta assuma o novo perfil. Salve a política em um arquivo chamado trust-policy.json.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

3. Crie um documento de política do IAM, conforme mostrado abaixo. Salve a política em um arquivo chamado policy.json.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "elasticache:Connect"  
      ],  
      "Resource": [  
        "arn:aws:elasticache:us-east-1:123456789012:serverless-cache:cache-01"  
      ]  
    }  
  ]  
}
```

```
"arn:aws:elasticache:us-  
east-1:123456789012:serverlesscache:cache-01",  
  "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"  
    ]  
  }  
]  
}
```

4. Criar um perfil do IAM.

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

5. Crie a política do IAM.

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

6. Anexe a política do IAM à função.

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

7. Crie um novo usuário habilitado para o IAM.

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

8. Crie um grupo de usuários e anexe o usuário.

```
aws elasticache create-user-group \  
--user-group-id iam-user-group-01 \  
--engine redis \  
--user-ids default iam-user-01  
  
aws elasticache modify-serverless-cache \  

```

```
--serverless-cache-name cache-01 \
--user-group-id iam-user-group-01
```

Conexão

Conectar com token como senha

Primeiro, é necessário gerar o token de autenticação do IAM de curta duração usando uma [solicitação pré-assinada do AWS SigV4](#). Depois disso, você fornecerá o token de autenticação do IAM como senha ao se conectar a um cache do Valkey ou Redis OSS, conforme mostrado no exemplo abaixo.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
    DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for ElastiCache
// with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);
String iamAuthToken =
    iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(userId, iamAuthToken)
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Veja abaixo a definição de IAMAuthTokenRequest.

```
public class IAMAuthTokenRequest {
    private static final HttpMethodName REQUEST_METHOD = HttpMethodName.GET;
    private static final String REQUEST_PROTOCOL = "http://";
    private static final String PARAM_ACTION = "Action";
    private static final String PARAM_USER = "User";
    private static final String PARAM_RESOURCE_TYPE = "ResourceType";
    private static final String RESOURCE_TYPE_SERVERLESS_CACHE = "ServerlessCache";
    private static final String ACTION_NAME = "connect";
    private static final String SERVICE_NAME = "elasticache";
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final String userId;
    private final String cacheName;
    private final String region;
    private final boolean isServerless;

    public IAMAuthTokenRequest(String userId, String cacheName, String region, boolean
isServerless) {
        this.userId = userId;
        this.cacheName = cacheName;
        this.region = region;
        this.isServerless = isServerless;
    }

    public String toSignedRequestUri(AWSCredentials credentials) throws
URISyntaxException {
        Request<Void> request = getSignableRequest();
        sign(request, credentials);
        return new URIBuilder(request.getEndpoint())
            .addParameters(toNamedValuePair(request.getParameters()))
            .build()
            .toString()
            .replace(REQUEST_PROTOCOL, "");
    }

    private <T> Request<T> getSignableRequest() {
        Request<T> request = new DefaultRequest<>(SERVICE_NAME);
        request.setHttpMethod(REQUEST_METHOD);
        request.setEndpoint(getRequestUri());
        request.addParameters(PARAM_ACTION, Collections.singletonList(ACTION_NAME));
        request.addParameters(PARAM_USER, Collections.singletonList(userId));
        if (isServerless) {
```

```

        request.addParameters(PARAM_RESOURCE_TYPE,
Collections.singletonList(RESOURCE_TYPE_SERVERLESS_CACHE));
    }
    return request;
}

private URI getRequestUri() {
    return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, cacheName));
}

private <T> void sign(SignableRequest<T> request, AWSCredentials credentials) {
    AWS4Signer signer = new AWS4Signer();
    signer.setRegionName(region);
    signer.setServiceName(SERVICE_NAME);

    DateTime dateTime = DateTime.now();
    dateTime = dateTime.plus(Duration.standardSeconds(TOKEN_EXPIRY_SECONDS));

    signer.presignRequest(request, credentials, dateTime.toDate());
}

private static List<NameValuePair> toNamedValuePair(Map<String, List<String>> in) {
    return in.entrySet().stream()
        .map(e -> new BasicNameValuePair(e.getKey(), e.getValue().get(0)))
        .collect(Collectors.toList());
}
}

```

Conectar com o provedor de credenciais

O código abaixo mostra como se autenticar ElastiCache usando o provedor de credenciais de autenticação do IAM.

```

String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
    DefaultAWSCredentialsProviderChain();

```

```
// Create an IAM authentication token request. Once this request is signed it can be
// used as an
// IAM authentication token for ElastiCache with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);

// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
    RedisIAMAuthCredentialsProvider(
        userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Abaixo está um exemplo de um cliente Lettuce Redis OSS que envolve um provedor de credenciais para gerar IAMAuth TokenRequest automaticamente credenciais temporárias quando necessário.

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userId;
    private final IAMAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAuthCredentialsProvider(String userId,
        IAMAuthTokenRequest iamAuthTokenRequest,
        AWSCredentialsProvider awsCredentialsProvider) {
        this.userName = userName;
        this.awsCredentialsProvider = awsCredentialsProvider;
        this.iamAuthTokenRequest = iamAuthTokenRequest;
    }
}
```



```
        this.iamAuthTokenSupplier =  
        Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,  
        TimeUnit.SECONDS);  
    }  
  
    @Override  
    public Mono<RedisCredentials> resolveCredentials() {  
        return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));  
    }  
  
    private String getIamAuthToken() {  
        return  
        iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());  
    }  
}
```

Autenticar com o comando AUTH do Valkey e Redis OSS

Note

O AUTH foi substituído por [the section called “Regras de controle de acesso com base em função \(RBAC\)”](#). Todos os caches sem servidor devem usar o RBAC na autenticação.

As senhas ou tokens de autenticação do Valkey e Redis OSS habilitam o Valkey e Redis OSS a exigir uma senha antes de permitir que os clientes executem comandos, melhorando assim a segurança dos dados. O AUTH está disponível somente para clusters baseados em nós.

Tópicos

- [Visão geral do AUTH no ElastiCache Valkey e no Redis OSS](#)
- [Aplicando autenticação a um cluster OSS ElastiCache para Valkey e Redis](#)
- [Modificação do token AUTH em um cluster existente](#)
- [Migração do RBAC para o AUTH](#)

Visão geral do AUTH no ElastiCache Valkey e no Redis OSS

Quando você usa o AUTH com seu cluster OSS ElastiCache para Valkey e Redis, há alguns refinamentos.

Em particular, esteja ciente destas restrições de token ou senha AUTH ao usar AUTH:

- Tokens, ou senhas, devem ter de 16 a 128 caracteres imprimíveis.
- Caracteres não alfanuméricos são restritos a (!, &, #, \$, ^, <, >, -).
- O AUTH só pode ser habilitado para criptografia em trânsito para clusters do Valkey ou Redis OSS.

Para configurar um token forte, recomendamos que você siga uma política de senha estrita, tal como exigir o seguinte:

- Tokens ou senhas devem incluir pelo menos três dos seguintes tipos de caractere:
 - Caracteres maiúsculos
 - Caracteres minúsculos
 - Dígitos
 - Caracteres não alfanuméricos (!, &, #, \$, ^, <, >, -)
- Tokens ou senhas não devem conter uma palavra de dicionário ou uma palavra de dicionário minimamente alterada.
- Tokens ou senhas não devem ser iguais ou similares a um token usado recentemente.

Aplicando autenticação a um cluster OSS ElastiCache para Valkey e Redis

Você pode exigir que os usuários insiram um token (senha) em um servidor Valkey ou Redis OSS protegido por tokens. Para fazer isso, inclua o parâmetro `--auth-token` (API: `AuthToken`) com o token correto quando você criar seu grupo de replicação ou cluster. Inclua-o também em todos os comandos subsequentes para o grupo de replicação ou cluster.

A AWS CLI operação a seguir cria um grupo de replicação com a criptografia em trânsito (TLS) ativada e o AUTH token. ***This-is-a-sample-token*** Substitua o grupo de sub-redes `sng-test` por um grupo de sub-redes que exista.

Principais parâmetros

- **--engine**: precisa ser `valkey` ou `redis`.
- **--engine-version**: se o mecanismo for Redis OSS, é preciso que seja 3.2.6, 4.0.10 ou posterior.
- **--transit-encryption-enabled**: obrigatório para autenticação e qualificação para HIPAA.
- **--auth-token**: obrigatório para qualificação para HIPAA. Esse valor deve ser o token correto para esse servidor Valkey ou Redis OSS protegido por tokens.

- **--cache-subnet-group**: obrigatório para qualificação para HIPAA.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id authtestgroup \  
  --replication-group-description authtest \  
  --engine redis \  
  --cache-node-type cache.m4.large \  
  --num-node-groups 1 \  
  --replicas-per-node-group 2 \  
  --transit-encryption-enabled \  
  --auth-token This-is-a-sample-token \  
  --cache-subnet-group sng-test
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id authtestgroup ^  
  --replication-group-description authtest ^  
  --engine redis ^  
  --cache-node-type cache.m4.large ^  
  --num-node-groups 1 ^  
  --replicas-per-node-group 2 ^  
  --transit-encryption-enabled ^  
  --auth-token This-is-a-sample-token ^  
  --cache-subnet-group sng-test
```

Modificação do token AUTH em um cluster existente

Para facilitar a atualização da autenticação, você pode modificar o token AUTH usado em um cluster. Você pode fazer essa modificação se a versão do mecanismo for Valkey 7.2 ou superior ou Redis 5.0.6 ou superior. ElastiCache também deve ter a criptografia em trânsito ativada.

A modificação do token de autenticação oferece suporte a duas estratégias: ROTATE (ALTERNAR) e SET (DEFINIR). A estratégia ROTATE (ALTERNAR) acrescenta um token AUTH adicional para o servidor enquanto retém o token anterior. A estratégia SET (DEFINIR) atualiza o servidor para aceitar apenas um único token AUTH. Faça essas chamadas de modificação com o parâmetro `--apply-immediately` para aplicar as alterações imediatamente.

Alternância do token AUTH

Para atualizar um servidor Valkey ou Redis OSS com um novo token AUTH, chame a API `ModifyReplicationGroup` com o parâmetro `--auth-token` como o novo token AUTH e o `--auth-token-update-strategy` com o valor `ROTATE` (ALTERNAR). Depois que a modificação `ROTATE` (ALTERNAR) for concluída, o cluster oferecerá suporte ao token AUTH anterior além do especificado no parâmetro `auth-token`. Se nenhum token AUTH tiver sido configurado no grupo de replicação antes da rotação do token AUTH, o cluster oferecerá suporte ao token AUTH especificado no parâmetro `--auth-token`, além do suporte a conexões sem autenticação. Consulte [Definição do token AUTH](#) para atualizar o token AUTH a ser necessário usando a estratégia de atualização `SET` (DEFINIR).

Note

Se você não configurar o token AUTH antes, depois que a modificação for concluída, o cluster não oferecerá suporte a nenhum token AUTH além do especificado no parâmetro `auth-token`.

Se essa modificação for executada em um servidor que já ofereça suporte a dois tokens AUTH, o token AUTH mais antigo também será removido durante esta operação. Isso permite que um servidor ofereça suporte a até dois tokens AUTH mais recentes em um determinado momento.

Neste ponto, você pode continuar atualizando o cliente para usar o token AUTH mais recente. Depois que os clientes forem atualizados, você poderá usar a estratégia `SET` (DEFINIR) para alternância de token AUTH (explicada na seção a seguir) para começar exclusivamente a usar o novo token.

A AWS CLI operação a seguir modifica um grupo de replicação para girar o token. AUTH *This-is-the-rotated-token*

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-rotated-token \  
--auth-token-update-strategy ROTATE \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-rotated-token ^  
--auth-token-update-strategy ROTATE ^  
--apply-immediately
```

Definição do token AUTH

Para atualizar um servidor Valkey ou Redis OSS para oferecer suporte a um único token AUTH necessário, chame a operação `ModifyReplicationGroup` da API com o parâmetro `--auth-token` com o mesmo valor do último token AUTH e o parâmetro `--auth-token-update-strategy` com o valor `SET`. A estratégia `SET` (DEFINIR) só pode ser usada com um cluster que tenha 2 tokens AUTH ou 1 token AUTH opcional do uso anterior da estratégia `ROTATE` (ALTERNAR). Após a conclusão da modificação, o servidor oferecerá suporte apenas para o token AUTH especificado no parâmetro `auth-token`.

A AWS CLI operação a seguir modifica um grupo de replicação para definir o token AUTH. *This-is-the-set-token*

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-set-token \  
--auth-token-update-strategy SET \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-set-token ^  
--auth-token-update-strategy SET ^  
--apply-immediately
```

Habilitação de autenticação em um cluster existente

Para habilitar a autenticação em um servidor Valkey ou Redis OSS existente, chame a operação `ModifyReplicationGroup` da API. Execute `ModifyReplicationGroup` com o parâmetro `--`

auth-token como o novo token e o parâmetro `--auth-token-update-strategy` com o valor ROTATE (ALTERNAR).

Depois que a modificação ROTATE (ALTERNAR) for concluída, o cluster oferecerá suporte ao token AUTH especificado no parâmetro `--auth-token`, além do suporte a conexões sem autenticação. Depois que todos os aplicativos cliente forem atualizados para se autenticar no Valkey ou no Redis OSS com o token AUTH, use a estratégia SET (DEFINIR) para marcar o token AUTH conforme necessário. A habilitação da autenticação só é aceita em servidores Valkey e Redis OSS com criptografia em trânsito (TLS) habilitada.

Migração do RBAC para o AUTH

Se você estiver autenticando usuários com o controle de acesso baseado em função (RBAC) do Valkey ou Redis OSS, conforme descrito em [Regras de controle de acesso com base em função \(RBAC\)](#), e desejar migrar para o AUTH, use os seguintes procedimentos. Você pode migrar usando o console ou a CLI.

Para migrar do RBAC para o AUTH usando o console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região em que o cluster que você deseja modificar está localizado.
3. No painel de navegação, escolha o mecanismo em execução no cluster que deseja modificar.

É exibida uma lista dos clusters do mecanismo escolhido.

4. Na lista de clusters, no cluster que você deseja modificar, escolha seu nome.
5. Para Actions (Ações), escolha Modify (Modificar).

A janela Modificar é exibida.

6. Em Controle de acesso, escolha Acesso do usuário padrão AUTH do Valkey ou Acesso do usuário padrão AUTH do Redis OSS.
7. Em Token AUTH do Valkey ou Token AUTH do Redis OSS, defina um novo token.
8. Escolha Previsualizar alterações e, na próxima tela, Modificar.

Para migrar do RBAC para o AUTH usando o AWS CLI

Use um dos comandos a seguir para configurar um novo token AUTH opcional para seu grupo de replicação do Valkey ou Redis OSS. Observe que um token Auth opcional permitirá acesso não autenticado ao grupo de replicação até que o token Auth seja marcado como obrigatório, usando a estratégia de atualização SET na etapa a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test \  
  --remove-user-groups \  
  --auth-token This-is-a-sample-token \  
  --auth-token-update-strategy ROTATE \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test ^  
  --remove-user-groups ^  
  --auth-token This-is-a-sample-token ^  
  --auth-token-update-strategy ROTATE ^  
  --apply-immediately
```

Depois de executar o comando acima, você pode atualizar seus aplicativos Valkey ou Redis OSS para se autenticar no grupo de ElastiCache replicação usando o token AUTH opcional recém-configurado. Para concluir a rotação do token Auth, use a estratégia de atualização SET no comando subsequente abaixo. Isso marcará o token AUTH opcional conforme necessário. Quando a atualização do token Auth for concluída, o status do grupo de replicação será exibido como ACTIVE e todas as conexões com esse grupo de replicação exigirão autenticação.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test \  
  --auth-token This-is-a-sample-token \  
  --auth-token-update-strategy SET \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^
```

```
--replication-group-id test ^
--remove-user-groups ^
--auth-token This-is-a-sample-token ^
--auth-token-update-strategy SET ^
--apply-immediately
```

Para obter mais informações, consulte [Autenticar com o comando AUTH do Valkey e Redis OSS](#).

 Note

Se você precisar desativar o controle de acesso em um ElastiCache cluster, consulte [the section called “Desabilitar o controle de acesso em um cache do ElastiCache Valkey ou Redis OSS”](#).

Desabilitar o controle de acesso em um cache do ElastiCache Valkey ou Redis OSS

Siga as instruções abaixo para desabilitar o controle de acesso em um cache do Valkey ou Redis OSS habilitado para TLS. Seu cache terá um dos dois tipos diferentes de configurações: acesso de usuário padrão AUTH ou lista de controle de acesso de grupo de usuários (RBAC). Se o cache tiver sido criado com a configuração AUTH, você precisará alterá-lo para a configuração RBAC antes de poder desabilitar o cache por meio da remoção dos grupos de usuários. Se o cache tiver sido criado com a configuração RBAC, você poderá desabilitá-lo diretamente.

Como desabilitar um cache de tecnologia sem servidor do Valkey ou Redis OSS configurado com RBAC

1. Remova os grupos de usuários para desabilitar o controle de acesso.

```
aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>
--remove-user-group
```

2. (Opcional) Verifique se nenhum grupo de usuários está associado ao cache sem servidor.

```
aws elasticache describe-serverless-caches --serverless-cache-name <serverless-
cache>
{
    "...
    "UserGroupId": ""
    "...
}
```



```
}
```

Como desabilitar um cache do Valkey ou Redis OSS configurado com um token AUTH

1. Altere o token AUTH para RBAC e especifique um grupo de usuários a ser adicionado.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-value>
```

2. Verifique se o token AUTH foi desabilitado e se um grupo de usuários foi adicionado.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-id-value>
{
  "...",
  "AuthTokenEnabled": false,
  "UserGroupIds": [
    "<user-group-value>"
  ]
  "...",
}
```

3. Remova os grupos de usuários para desabilitar o controle de acesso.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
  "...",
  "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
  },
  "...",
}
```

4. (Opcional) Verifique se nenhum grupo de usuários está associado ao cluster. O campo `AuthTokenEnabled` também deve ser falso.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-value>
"AuthTokenEnabled": false
```

Como desabilitar um cluster do Valkey ou Redis OSS configurado com RBAC

1. Remova os grupos de usuários para desabilitar o controle de acesso.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
  "...":
    "PendingModifiedValues": {
      "UserGroups": {
        "UserGroupIdsToAdd": [],
        "UserGroupIdsToRemove": [
          "<user-group-value>"
        ]
      }
    }
  "...":
}
```

2. (Opcional) Verifique se nenhum grupo de usuários está associado ao cluster. O campo AuthTokenEnabled também deve ser falso.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-value>
"AuthTokenEnabled": false
```

Privacidade do tráfego entre redes

O Amazon ElastiCache usa as seguintes técnicas para proteger seus dados de cache e protegê-los contra o acesso não autorizado:

- [Amazon VPCs e ElastiCache segurança](#) explica o tipo de grupo de segurança de que você precisa para sua instalação.
- [Identity and Access Management para Amazon ElastiCache](#) para conceder e limitar ações de usuários, grupos e funções.

Tópicos

- [Amazon VPCs e ElastiCache segurança](#)
- [ElastiCache API e interface VPC endpoints \(AWS PrivateLink\)](#)
- [Sub-redes e grupos de sub-redes](#)

Amazon VPCs e ElastiCache segurança

Como a segurança dos dados é importante, o ElastiCache fornece meios para você controlar quem tem acesso aos seus dados. A forma como você controla o acesso aos seus dados depende de você ter lançado ou não seus clusters em uma Amazon Virtual Private Cloud (Amazon VPC) ou Amazon EC2 -Classic.

Important

Nós descontinuamos o uso do Amazon EC2 -Classic para lançar clusters. ElastiCache Todos os nós de geração atuais são iniciados somente na Amazon Virtual Private Cloud.

O serviço da Amazon Virtual Private Cloud (Amazon VPC) define uma rede virtual que lembra muito um datacenter tradicional. Ao configurar sua Amazon VPC, você pode selecionar o intervalo de endereços IP dela, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações de segurança. Você também pode adicionar um cluster à rede virtual e controlar o acesso ao cluster usando os grupos de segurança da Amazon VPC.

Esta seção explica como configurar manualmente um ElastiCache cluster em uma Amazon VPC. Essas informações são destinadas a usuários que desejam uma compreensão mais profunda de como ElastiCache e a Amazon VPC trabalham juntas.

Tópicos

- [Entendendo o ElastiCache e as Amazon VPCs](#)
- [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#)
- [Criar uma nuvem privada virtual \(VPC\)](#)
- [Conexão a um cache em execução em um Amazon VPC](#)

Entendendo o ElastiCache e as Amazon VPCs

O ElastiCache está totalmente integrado à Amazon Virtual Private Cloud (Amazon VPC). Para usuários do ElastiCache, isso significa o seguinte:

- Se a sua conta da AWS oferece suporte apenas para a plataforma EC2-VPC, o ElastiCache sempre inicia seu cluster em uma Amazon VPC.
- Se você for iniciante na AWS, seus clusters serão implantados em uma Amazon VPC. Uma VPC padrão será criada para você automaticamente.
- Se você tiver uma VPC padrão e não especificar uma sub-rede quando executar um cluster, este será iniciado na sua Amazon VPC padrão.

Para mais informações, consulte [Detecção de suas plataformas compatíveis e se você tem um VPC padrão](#).

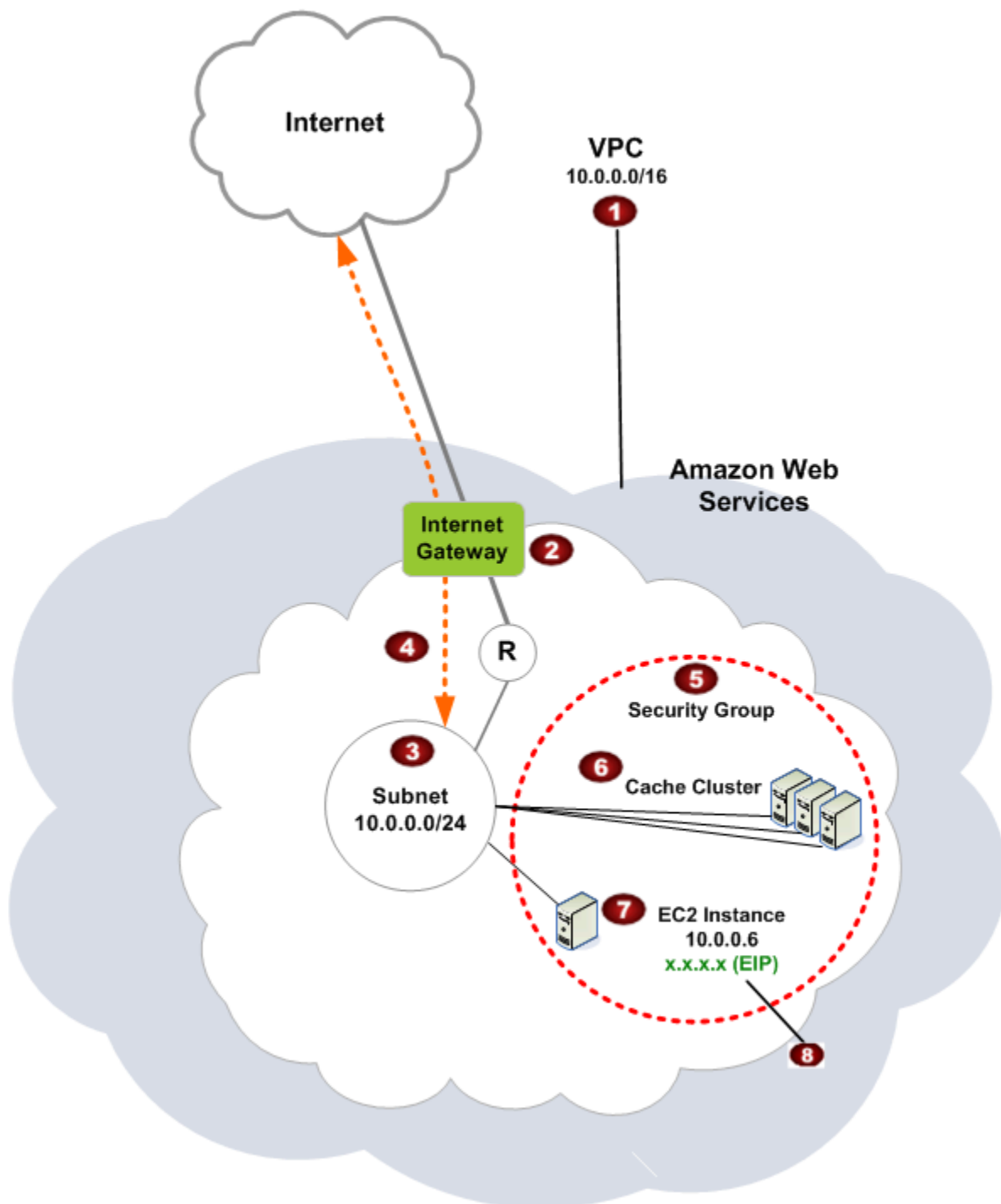
Com a Amazon Virtual Private Cloud, você pode criar uma rede virtual na nuvem da AWS que se assemelha muito com um datacenter tradicional. É possível configurar sua Amazon VPC, incluindo selecionar o intervalo de endereços IP, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações de segurança.

A funcionalidade básica do ElastiCache é a mesma em uma nuvem privada virtual. O ElastiCache gerencia atualizações de software, a aplicação de patch, a detecção de falhas e a recuperação, independentemente de os seus clusters estarem implantados dentro ou fora de uma Amazon VPC.

Os nós de cache do ElastiCache implantados fora de uma Amazon VPC recebem um endereço IP para o qual o endpoint/nome do DNS é resolvido. Isso fornece conectividade entre as instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Quando você ativa um cluster do ElastiCache na sub-rede privada de uma Amazon VPC, cada nó de cache recebe um endereço IP privado dentro dessa sub-rede.

Visão geral do ElastiCache em uma Amazon VPC

O seguinte diagrama e tabela descrevem o ambiente da Amazon VPC, juntamente com os clusters do ElastiCache e as instâncias do Amazon EC2 que são executadas na Amazon VPC.

**1**

A Amazon VPC é uma parte isolada da nuvem da AWS que recebe seu próprio bloco de endereços IP.

2

Um gateway da Internet conecta sua Amazon VPC diretamente à Internet e fornece acesso a outros recursos da AWS, como o Amazon Simple Storage Service (Amazon S3), que estão em execução fora da sua Amazon VPC.

3

Uma sub-rede da Amazon VPC é um segmento do intervalo de endereços IP de uma Amazon VPC em que você pode isolar recursos da AWS de acordo com suas necessidades de segurança e operacionais.

4

Uma tabela de roteamento na Amazon VPC direciona o tráfego de rede entre a sub-rede e a Internet. A Amazon VPC tem um roteador implícito, que é simbolizado neste diagrama pelo círculo com o R.

5

Um grupo de segurança da Amazon VPC controla o tráfego de entrada e saída para seus clusters do ElastiCache e suas instâncias do Amazon EC2.

6

Você pode ativar um cluster do ElastiCache na sub-rede. Os nós de cache possuem endereços IP privados a partir do intervalo de endereços da sub-rede.

7

Você também pode ativar instâncias do Amazon EC2 na sub-rede. Cada instância do Amazon EC2 tem um endereço IP privado do intervalo de endereços da sub-rede. A instância do Amazon EC2 pode se conectar a qualquer nó de cache na mesma sub-rede.

8

Para que uma instância do Amazon EC2 na sua Amazon VPC seja acessível na Internet, você precisa atribuir um endereço público estático chamado de endereço IP elástico à instância.

Pré-requisitos

Para criar um cluster do ElastiCache dentro de uma Amazon VPC, sua Amazon VPC deve atender aos seguintes requisitos:

- A Amazon VPC deve permitir instâncias do Amazon EC2 não dedicadas. Você não pode usar o ElastiCache em uma Amazon VPC que está configurada para a locação de instâncias dedicadas.

- Um grupo de sub-redes de cache deve ser definido para a sua Amazon VPC. O ElastiCache usa esse grupo de sub-redes de cache para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos endpoints da VPC ou aos nós de cache.
- Os blocos CIDR para cada sub-rede devem ser suficientemente grandes para fornecer endereços IP de reposição para o ElastiCache usar durante atividades de manutenção.

Roteamento e segurança

Você pode configurar o roteamento na sua Amazon VPC para controlar para onde o tráfego flui (por exemplo, para o gateway da Internet ou o virtual private gateway). Com um gateway da Internet, sua Amazon VPC tem acesso direto a outros recursos da AWS que não estão sendo executados na sua Amazon VPC. Se você optar por ter apenas um gateway privado virtual com uma conexão com a rede local da sua organização, poderá rotear seu tráfego vinculado à Internet através da VPN e usar políticas de segurança locais e um firewall para controlar a saída. Nesse caso, você está sujeito a cobranças adicionais de largura de banda ao acessar os recursos da AWS pela Internet.

Você pode usar grupos de segurança do Amazon VPC para ajudar a proteger os clusters do ElastiCache e as instâncias do Amazon EC2 na sua Amazon VPC. Os security groups atuam como um firewall no nível da instância e não no nível da sub-rede.

Note

Recomendamos enfaticamente que você use nomes DNS para se conectar aos seus nós, pois o endereço IP subjacente pode mudar.

Documentação da Amazon VPC

A Amazon VPC tem seu próprio conjunto de documentação para descrever como criar e usar sua Amazon VPC. A tabela a seguir fornece links para guias sobre Amazon VPCs.

Descrição	Documentação
Como começar a usar a Amazon VPC	Conceitos básicos do Amazon VPC
Como usar a Amazon VPC por meio do Console de gerenciamento da AWS	Guia do usuário da Amazon VPC

Descrição	Documentação
Descrições completas de todos os comandos da Amazon VPC	Referência da linha de comando do Amazon EC2 (os comandos da Amazon VPC são encontrados na referência do Amazon EC2)
Descrições completas das operações da API da Amazon VPC, tipos de dados e erros da Amazon VPC	Referência da linha de comando do Amazon EC2 (as operações da API do Amazon VPC são encontrados na referência do Amazon EC2)
Informações para o administrador da rede que precisa configurar o gateway no final de uma conexão VPN IPsec opcional	O que é a AWS VPN Site-to-Site?

Para obter informações mais detalhadas sobre a Amazon Virtual Private Cloud, consulte [Amazon Virtual Private Cloud](#).

Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC

O Amazon ElastiCache oferece suporte para os seguintes cenários para acessar um cache em um Amazon VPC:

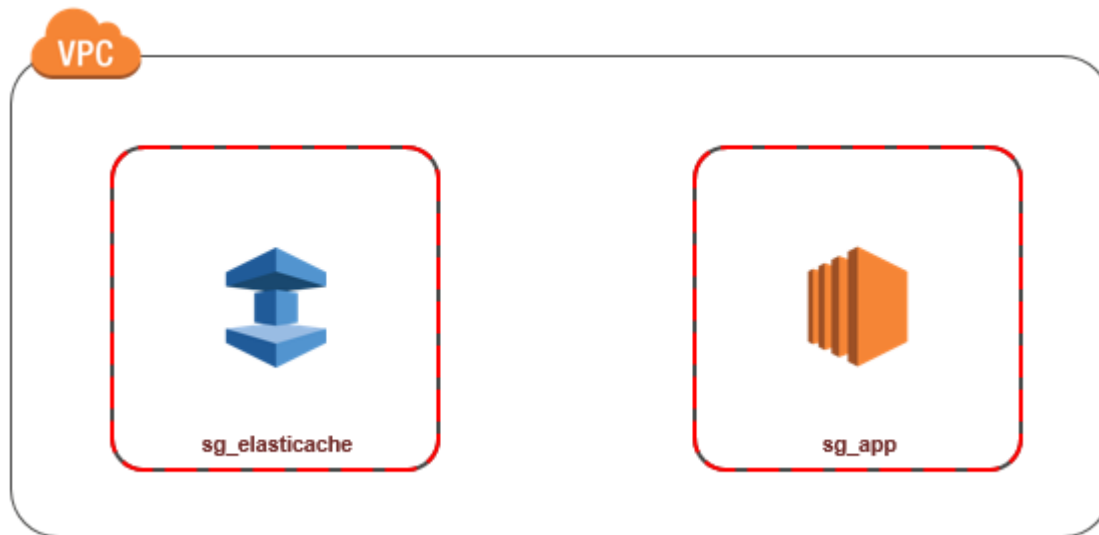
Sumário

- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão no mesmo Amazon VPC](#)
- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs](#)
 - [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs na mesma região](#)
 - [Uso do Transit Gateway](#)
 - [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs em regiões diferentes](#)
 - [Uso da VPC de trânsito](#)
- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente](#)
 - [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando conectividade de VPN](#)
 - [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando o Direct Connect](#)

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão no mesmo Amazon VPC

O caso de uso mais comum é quando uma aplicação implantada em uma instância do EC2 precisa se conectar a um cache na mesma VPC.

O diagrama a seguir ilustra esse cenário.



A maneira mais simples de gerenciar o acesso entre instâncias do EC2 e caches na mesma VPC é fazer o seguinte:

1. Crie um grupo de segurança da VPC para o cache. Esse grupo de segurança pode ser usado para restringir o acesso ao cache. Por exemplo, é possível criar uma regra personalizada para esse grupo de segurança que permite o acesso TCP usando a porta atribuída ao cache quando você o criou e um endereço IP que será usado para acessar o cache.

A porta padrão dos caches Memcached é 11211.

A porta padrão para caches do Valkey e do Redis OSS é 6379.

2. Crie um grupo de segurança de VPC para suas instâncias do EC2 (servidores Web e de aplicativos). Esse grupo de segurança pode, se necessário, permitir o acesso à instância do EC2 da Internet através da tabela de rotas da VPC. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o acesso TCP à instância do EC2 pela porta 22.
3. Crie regras personalizadas no grupo de segurança para o seu cache que permitam conexões do grupo de segurança que você criou para suas instâncias do EC2. Isso permitiria que qualquer membro de grupo de segurança acessasse os caches.

Note

Se você estiver planejando usar [Zonas locais](#), verifique se você as habilitou. Quando você criar um grupo de sub-redes nessa região local, sua VPC será estendida para essa zona local e sua VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona

de disponibilidade. Todos os gateways e tabelas de rotas relevantes serão ajustados automaticamente.

Para criar uma regra em um grupo de segurança de VPC que permita conexões de outro grupo de segurança

1. Faça login no Console de Gerenciamento da AWS e abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc>.
2. No painel de navegação, escolha Security Groups (Grupos de segurança).
3. Selecione ou crie um grupo de segurança que você usará para seus caches. Em Regras de entrada, selecione Editar regras de entrada e escolha Adicionar regra. Esse grupo de segurança permitirá o acesso a membros de outro grupo de segurança.
4. Em Tipo, escolha Regra TCP personalizada.
 - a. Para Port Range, especifique a porta que você usou quando criou seu cache.

A porta padrão dos caches Memcached é 11211.

A porta padrão dos caches e grupos de replicação do Valkey e do Redis OSS é 6379.
 - b. Na caixa Source, comece a digitar o ID do grupo de segurança. Na lista, selecione o grupo de segurança que você usará para o suas instâncias do Amazon EC2.
5. Escolha Save quando terminar.

Inbound rules (3)

Filter security group rules

Manage tags

Edit inbound rules

1

☐	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
☐	-	sg-	IPv4	SSH	TCP	22	0.0.0.0/0	-
☐	-	sg-	-	Custom TCP	TCP	6379	sg- / default	-

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs

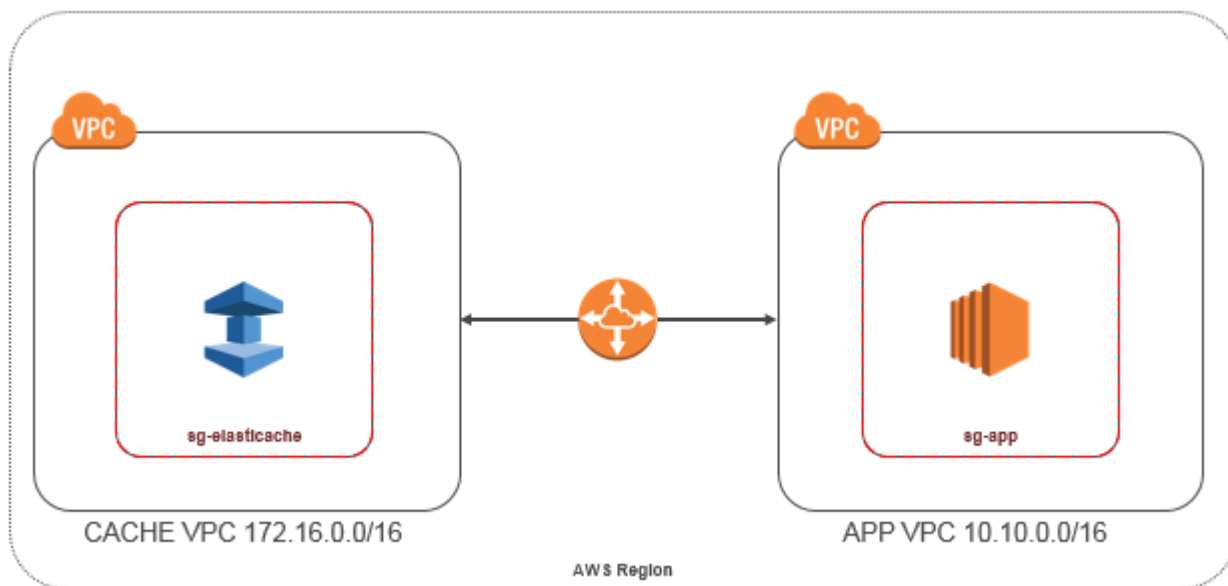
Quando seu cache está em uma VPC diferente da instância do EC2 que você está usando para acessá-la, existem várias maneiras de acessar o cache. Se o cache e a instância do EC2 estiverem em VPCs diferentes, mas na mesma região, você poderá usar o emparelhamento de VPCs. Se o cache e a instância do EC2 estiverem em regiões diferentes, você poderá criar conectividade via VPN entre regiões.

Tópicos

- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs na mesma região](#)
- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs em regiões diferentes](#)

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs na mesma região

O diagrama a seguir ilustra o acesso a um cache por uma instância do Amazon EC2 em um Amazon VPC diferente na mesma região usando uma conexão de emparelhamento do Amazon VPC.



Cache acessado por uma instância do Amazon EC2 em um Amazon VPC diferente na mesma região: conexão de emparelhamento da VPC

Uma conexão de emparelhamento da VPC é uma conexão de redes entre duas VPCs que permite direcionar o tráfego entre elas usando endereços IP privados. Instâncias em qualquer VPC podem se comunicar umas com as outras como se estivessem na mesma rede. Você pode criar uma conexão de emparelhamento de VPCs entre suas próprias Amazon VPCs ou com uma Amazon VPC em outra conta da AWS em uma única região. Para saber mais sobre o emparelhamento de Amazon VPCs, consulte a [documentação da VPC](#).

Note

A resolução de nomes DNS pode falhar para VPCs emparelhadas, dependendo das configurações aplicadas à VPC do ElastiCache. Para resolver isso, as duas VPCs devem ser habilitadas para nomes de hosts DNS e resolução DNS. Para obter mais informações, consulte [Habilitar a resolução de DNS para a conexão de emparelhamento da VPC](#).

Para acessar um cache em um Amazon VPC diferente por emparelhamento

1. Certifique-se de que as duas VPCs não tenham um intervalo de IP sobreposto, ou você não poderá compará-las.
2. Emparelhe as duas VPCs. Para obter mais informações, consulte [Criação e aceitação de uma conexão de emparelhamento da Amazon VPC](#).
3. Atualize sua tabela de roteamento. Para obter mais informações, consulte [Atualizar as tabelas de rotas para uma conexão de emparelhamento de VPC](#)

Veja a seguir a aparência das tabelas de rotas para o exemplo do diagrama anterior. Observe que pcx-a894f1c1 é a conexão de emparelhamento.

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

Tabela de roteamento de VPC

4. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada do grupo de segurança da aplicação na VPC emparelhada. Para obter mais informações, consulte a [Referência para security groups de VPC de emparelhamento](#).

O acesso a um cache por meio de uma conexão de emparelhamento implicará custos adicionais de transferência de dados.

Uso do Transit Gateway

Um Transit Gateway permite anexar VPCs e conexões VPN na mesma região da AWS e rotear tráfego entre elas. Um Transit Gateway funciona em contas da AWS, e você pode usar o Resource Access Manager da AWS para compartilhar o Transit Gateway com outras contas. Depois de compartilhar um gateway de trânsito com outra conta da AWS, o proprietário da conta poderá anexar as VPCs dele ao gateway de trânsito. Um usuário de qualquer uma das contas pode excluir o anexo a qualquer momento.

É possível ativar o multicast em um gateway de trânsito e, depois, criar um domínio de multicast do gateway de trânsito que permita ao tráfego de multicast ser enviado da origem de multicast para membros do grupo de multicast em anexos da VPC associados ao domínio.

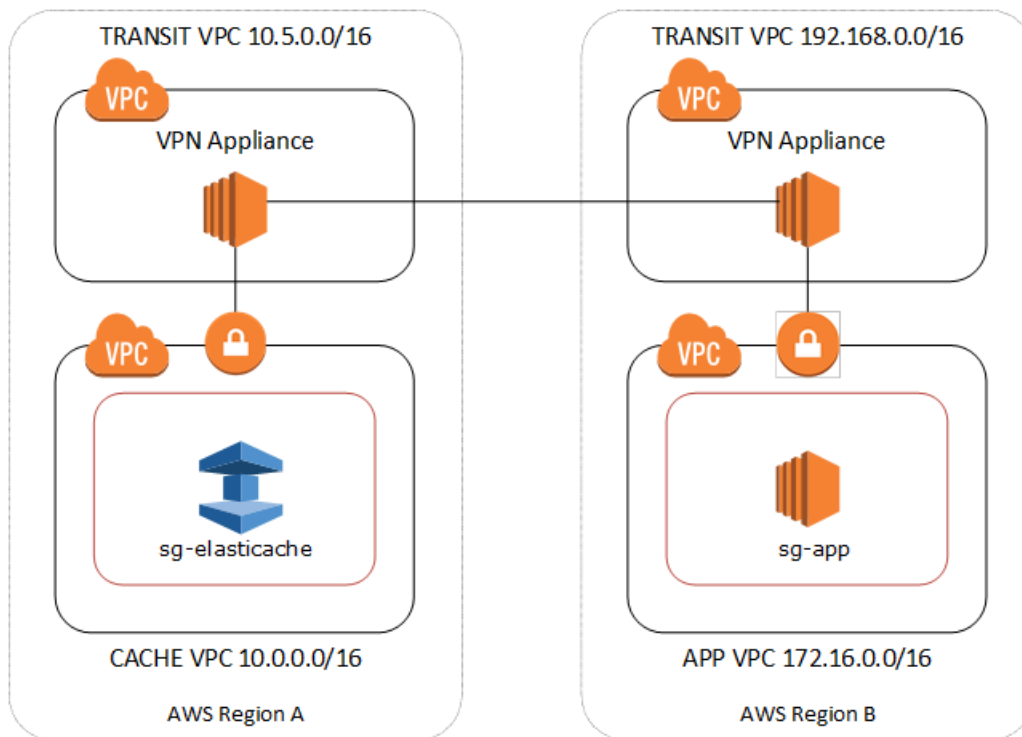
Também é possível criar um anexo da conexão de emparelhamento entre gateways de trânsito em diferentes regiões da AWS. Isso permite que você roteie o tráfego entre os anexos dos gateways de trânsito em regiões diferentes.

Para obter mais informações, consulte [Gateways de trânsito](#).

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs em regiões diferentes

Uso da VPC de trânsito

Uma alternativa ao uso do emparelhamento de VPC, outra estratégia comum para conectar várias VPCs geograficamente dispersas e redes remotas é criar uma VPC de trânsito que serve como um centro de trânsito de rede global. Uma VPC de trânsito simplifica o gerenciamento da rede e minimiza o número de conexões necessárias para conectar várias VPCs e redes remotas. Esse design pode economizar tempo e esforços e também reduzir custos, uma vez que é implementado praticamente sem as despesas tradicionais de estabelecer uma presença física em um hub de trânsito de colocação ou implantar equipamentos de rede física.



Conexão entre diferentes VPCs em regiões distintas

Depois de estabelecido o Amazon VPC de trânsito, uma aplicação implantada em uma VPC “spoke” em uma região pode se conectar a um cache do ElastiCache em uma VPC “spoke” em outra região.

Para acessar um cache em uma VPC diferente dentro de uma região da AWS diferente

1. Implante uma solução de VPC de trânsito. Para obter mais informações, consulte [Transit Gateway da AWS](#).
2. Atualize as tabelas de roteamento da VPC nas VPC de aplicativos e cache para rotear o tráfego através do VGW (gateway privado virtual) e do dispositivo de VPN. No caso do Roteamento dinâmico com o protocolo BGP, suas rotas podem ser propagadas automaticamente.
3. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada do intervalo IP de instâncias da aplicação. Observe que você não poderá fazer referência ao security group do servidor de aplicativos nesse cenário.

O acesso a um cache entre regiões introduzirá latências de rede e custos adicionais de transferência de dados entre regiões.

Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente

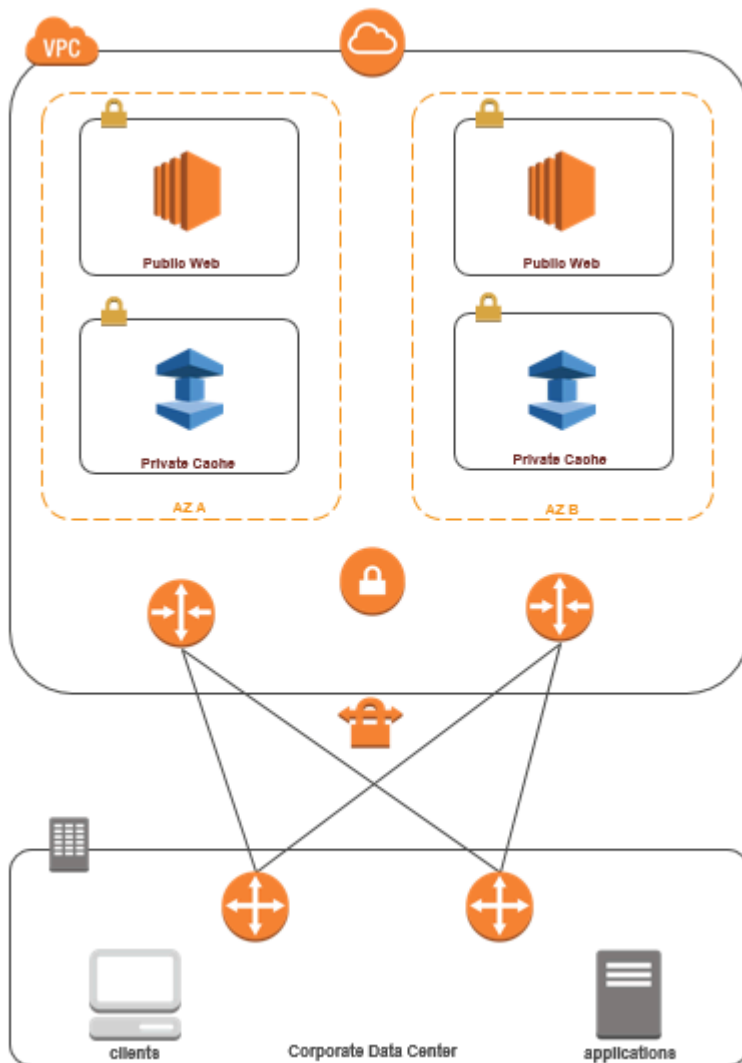
Outro cenário possível é uma arquitetura híbrida em que clientes ou aplicações no datacenter do cliente podem precisar acessar um cache do ElastiCache na VPC. Esse cenário também tem suporte, desde que haja conectividade entre a VPC dos clientes e o datacenter via VPN ou Direct Connect.

Tópicos

- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando conectividade de VPN](#)
- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando o Direct Connect](#)

Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando conectividade de VPN

O diagrama a seguir ilustra o acesso a um cache do ElastiCache a partir de uma aplicação executada na sua rede corporativa usando conexões VPN.



Conexão ao ElastiCache a partir do seu datacenter através de uma VPN

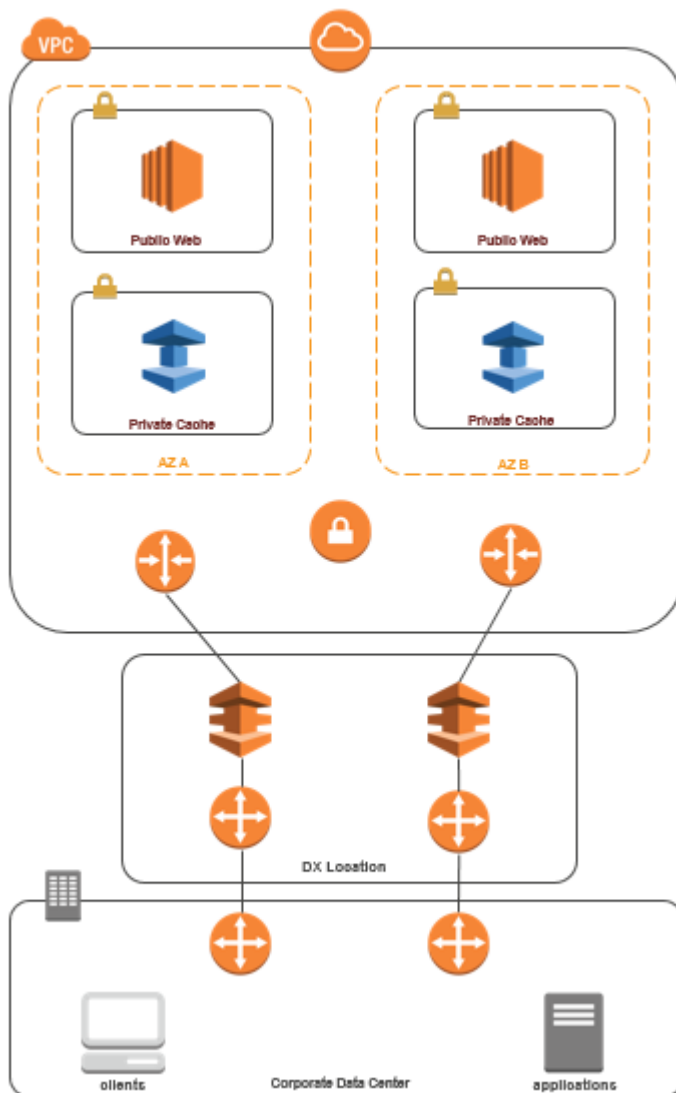
Para acessar um cache em uma VPC a partir do aplicativo no local via conexão VPN

1. Estabeleça a conectividade de VPN adicionando um gateway privado virtual de hardware à sua VPC. Para obter mais informações, consulte o tópico sobre como [Adicionar um gateway privado virtual de hardware à sua VPC](#).
2. Atualize a tabela de roteamento de VPC para a sub-rede na qual seu cache do ElastiCache está implantado para permitir o tráfego do seu servidor de aplicações on-premises. No caso do Roteamento dinâmico com o BGP, suas rotas podem ser propagadas automaticamente.
3. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada dos servidores de aplicações on-premises.

Acessar um cache através de uma conexão VPN introduzirá latências de rede e custos adicionais de transferência de dados.

Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando o Direct Connect

O diagrama a seguir ilustra o acesso a um cache do ElastiCache a partir de uma aplicação executada na sua rede corporativa usando o Direct Connect.



Conexão ao ElastiCache a partir do seu datacenter via Direct Connect

Para acessar um cache do ElastiCache de uma aplicação executada em sua rede usando o Direct Connect

1. Estabeleça a conectividade Direct Connect. Para obter mais informações, consulte [Conceitos básicos do Direct Connect da AWS](#).
2. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada dos servidores de aplicações on-premises.

O acesso a um cache por meio de uma conexão DX pode introduzir latências de rede e taxas adicionais de transferência de dados.

Criar uma nuvem privada virtual (VPC)

Neste exemplo, você cria uma Amazon VPC com uma sub-rede privada para cada zona de disponibilidade.

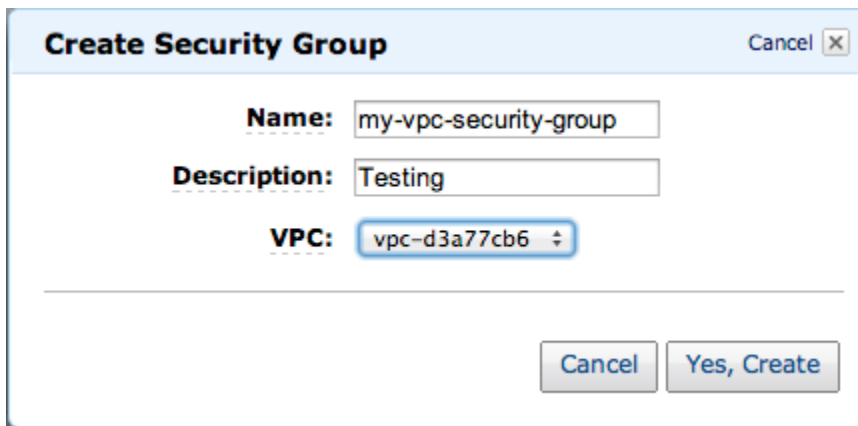
Criação de uma Amazon VPC (console)

1. Faça login no AWS Management Console e abra o console da Amazon VPC em. <https://console.aws.amazon.com/vpc/>
2. No painel da VPC, escolha Criar VPC.
3. Em Recursos a serem criados, escolha VPC e mais.
4. Em Número de zonas de disponibilidade (AZs), escolha o número de zonas de disponibilidade nas quais você deseja iniciar suas sub-redes.
5. Em Número de sub-redes públicas, escolha o número de sub-redes públicas que você deseja adicionar à sua VPC.
6. Em Número de sub-redes privadas, escolha o número de sub-redes públicas que você deseja adicionar à sua VPC.

Tip

Anote os identificadores das sub-redes e indique quais são públicas e quais são privadas. Você precisará dessas informações mais tarde quando iniciar seus clusters e adicionar uma EC2 instância da Amazon à sua Amazon VPC.

7. Crie um grupo de segurança da Amazon VPC. Você usará esse grupo para seu cluster e sua EC2 instância da Amazon.
 - a. No painel de navegação do console de gerenciamento da Amazon VPC, escolha Security Groups (Grupos de segurança).
 - b. Escolha Criar grupo de segurança.
 - c. Digite um nome e uma descrição do seu security group nas caixas correspondentes. Na caixa VPC, escolha o identificador da sua Amazon VPC.



- d. Quando estiver satisfeito com as configurações, clique em Yes, Create.
8. Defina uma regra de entrada de rede para seu security group. Essa regra permitirá que você se conecte à sua EC2 instância da Amazon usando o Secure Shell (SSH).
 - a. Na lista de navegação, escolha Security Groups.
 - b. Localize seu security group na lista e escolha-o.
 - c. Em Security Group, escolha a guia Inbound. Na caixa Create a new rule, escolha SSH e depois Add Rule.
 - d. Defina os seguintes valores para a sua nova regra de entrada a fim de permitir o acesso HTTP.
 - Tipo: HTTP
 - Origem: 0.0.0.0/0

Escolha Apply Rule Changes.

Agora você está pronto para criar um grupo de sub-redes de cache e executar um cluster na sua Amazon VPC.

- [Criação de um grupo de sub-redes](#)
- [Criação de um cluster do Memcached \(console\).](#)
- [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\).](#)

Conexão a um cache em execução em um Amazon VPC

Este exemplo mostra como iniciar uma EC2 instância da Amazon em sua Amazon VPC. Em seguida, você pode fazer login nessa instância e acessar o ElastiCache cache que está sendo executado na Amazon VPC.

Conexão a um cache em execução em um Amazon VPC (console)

Neste exemplo, você cria uma EC2 instância da Amazon em sua Amazon VPC. Você pode usar essa EC2 instância da Amazon para se conectar aos nós de cache em execução na Amazon VPC.

Note

Para obter informações sobre o uso da Amazon EC2, consulte o [Amazon EC2 Getting Started Guide](#) na [EC2 documentação da Amazon](#).

Para criar uma EC2 instância da Amazon em sua Amazon VPC usando o console da Amazon EC2

1. Faça login no Console de gerenciamento da AWS e abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
2. No console, escolha Executar instância e siga estas etapas:
3. Na página Selecione uma Imagem de máquina da Amazon (AMI), escolha a AMI Amazon Linux de 64 bits e Selecionar.
4. Na página Choose an Instance Type (Escolher um tipo de instância), escolha 3. Configure Instance (Configurar instância).
5. Na página Configure os detalhes da instância, faça as seguintes seleções:
 - a. Na lista Network (Rede), escolha sua Amazon VPC.
 - b. Na lista Sub-rede, escolha sua sub-rede pública.

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 ⓘ 1

Purchasing option ⓘ ☐ Request Spot Instances

Network ⓘ vpc-d3a77cb6 (10.0.0.0/16) ⓘ Create new VPC

Subnet ⓘ subnet-58f5e63a(10.0.0.0/24) | sa-east-1a ⓘ Create new subnet
250 IP Addresses available

Public IP ⓘ ☐ Automatically assign a public IP address to your instances

Quando estiver satisfeito com as configurações, escolha 4. Add Storage (Adicionar armazenamento).

6. Na página Add Storage (Adicionar armazenamento), escolha 5. Tag Instance (Marcar a instância).
7. Na página Tag Instance, digite um nome para sua EC2 instância Amazon e escolha 6. Configure Security Group (Configurar o grupo de segurança).
8. Na página Configurar grupo de segurança, escolha Selecionar um grupo de segurança existente. Para obter mais informações sobre grupos de segurança, consulte [Grupos EC2 de segurança da Amazon para instâncias 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 your instance. For 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 ports. 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
☐ sg-1a3d2178	default	default VPC security group
☒ sg-f13d2193	my-vpc-security-group	Testing

Escolha o nome do seu grupo de segurança da Amazon VPC depois escolha Review and Launch (Analisar e iniciar).

9. Na página Revisar instância e executar, escolha Executar.
10. Na janela Select an existing key pair or create a new key pair, especifique um par de chaves que você deseja usar com essa instância.

 Note

Para obter informações sobre o gerenciamento de pares de chaves, consulte o [Amazon EC2 Getting Started Guide](#).

11. Quando você estiver pronto para iniciar sua EC2 instância da Amazon, escolha Launch.

Agora você pode atribuir um endereço IP elástico à EC2 instância da Amazon que você acabou de criar. Você precisa usar esse endereço IP para se conectar à EC2 instância da Amazon.

Como atribuir um endereço IP elástico (console)

1. Abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.
2. Na lista de navegação, escolha Elastic IPs.
3. Escolha Alocar endereço IP elástico.
4. Na caixa de diálogo Alocar endereço IP elástico, aceite o Grupo de borda de rede padrão e escolha Alocar.
5. Escolha o endereço IP elástico que você acabou de alocar na lista e escolha Associar endereço.
6. Na caixa de diálogo Associate Address, na caixa Instance, escolha o ID da EC2 instância Amazon que você iniciou.

Na caixa Endereço IP privado, selecione a caixa para obter o endereço IP privado e escolha Associar.

Agora você pode usar o SSH para se conectar à EC2 instância da Amazon usando o endereço IP elástico que você criou.

Para se conectar à sua EC2 instância Amazon

- Abra uma janela de comando. No prompt de comando, emita o comando a seguir, substituindo `mykeypair.pem` pelo nome do seu arquivo de par de chaves e `54.207.55.251` pelo seu endereço IP elástico.

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```


⚠ Important

Não saia da sua EC2 instância da Amazon ainda.

Agora você está pronto para interagir com seu ElastiCache cluster. Para poder fazer isso, caso ainda não o tenha feito, você precisa instalar o utilitário telnet.

Como instalar o telnet e interagir com seu cluster (AWS CLI)

- Abra uma janela de comando. No prompt de comando, emita o seguinte comando. No prompt de confirmação, digite 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!
```

Agora você pode se conectar a uma VPC com Memcached ou Redis.

Conectar-se a uma VPC com o Memcached

1. Acesse o ElastiCache console em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint de um dos nós do seu cluster. Para obter mais informações, consulte [Encontrar endpoints de conexão](#).

2. Use o telnet para se conectar ao endpoint do nó de cache pela porta 11211. Substitua o nome do host mostrado abaixo pelo nome do host do nó de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

Agora você está conectado ao mecanismo de cache e pode emitir comandos. Neste exemplo, você adiciona um item de dados ao cache e o obtém logo em seguida. Por último, você se desconectará do nó do cache.

Para armazenar uma chave e um valor, digite as duas linhas a seguir:

```
add mykey 0 3600 28  
This is the value for mykey
```

O mecanismo de cache responde com o seguinte:

```
OK
```

Para recuperar o valor para mykey, digite o seguinte:

```
get mykey
```

O mecanismo de cache responde com o seguinte:

```
VALUE mykey 0 28  
This is the value for my key  
END
```

Para desconectar-se do mecanismo de cache, digite o seguinte:

```
quit
```

Conectar-se a uma VPC com o Redis

1. Acesse o ElastiCache console em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint de um dos nós do seu cluster. Para obter mais informações, consulte [Encontrar endpoints de conexão](#) para Redis.

2. Use o telnet para se conectar ao endpoint do nó de cache pela porta 6379. Substitua o nome do host mostrado abaixo pelo nome do host do nó de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Agora você está conectado ao mecanismo de cache e pode emitir comandos. Neste exemplo, você adiciona um item de dados ao cache e o obtém logo em seguida. Por último, você se desconectará do nó do cache.

Para armazenar uma chave e um valor, digite as duas linhas a seguir:

```
set mykey myvalue
```

O mecanismo de cache responde com o seguinte:

```
OK
```

Para recuperar o valor para mykey, digite o seguinte:

```
get mykey
```

Para desconectar-se do mecanismo de cache, digite o seguinte:

```
quit
```

3. Acesse o ElastiCache console em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint de um dos nós do seu cluster. Para obter mais informações, consulte [Encontrar endpoints de conexão](#) para Redis OSS.
4. Use o telnet para se conectar ao endpoint do nó de cache pela porta 6379. Substitua o nome do host mostrado abaixo pelo nome do host do nó de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Agora você está conectado ao mecanismo de cache e pode emitir comandos. Neste exemplo, você adiciona um item de dados ao cache e o obtém logo em seguida. Por último, você se desconectará do nó do cache.

Para armazenar uma chave e um valor, digite o seguinte:

```
set mykey myvalue
```

O mecanismo de cache responde com o seguinte:

```
OK
```

Para recuperar o valor para mykey, digite o seguinte:

```
get mykey
```

O mecanismo de cache responde com o seguinte:

```
get mykey  
myvalue
```

Para desconectar-se do mecanismo de cache, digite o seguinte:

```
quit
```

Important

Para evitar cobranças adicionais em sua AWS conta, exclua todos AWS os recursos que você não deseja mais depois de testar esses exemplos.

ElastiCache API e interface VPC endpoints (AWS PrivateLink)

Você pode estabelecer uma conexão privada entre sua VPC e os endpoints de ElastiCache API da Amazon criando uma interface VPC endpoint. Os endpoints de interface são alimentados por [AWS PrivateLink](#). AWS PrivateLink permite que você acesse de forma privada as operações de ElastiCache API da Amazon sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão AWS Direct Connect.

As instâncias em sua VPC não precisam de endereços IP públicos para se comunicar com os endpoints de ElastiCache API da Amazon. Suas instâncias também não precisam de endereços IP

públicos para usar nenhuma das operações de ElastiCache API disponíveis. O tráfego entre sua VPC e a Amazon ElastiCache não sai da rede Amazon. Cada endpoint de interface é representado por uma ou mais interfaces de rede elástica nas sub-redes. Para obter mais informações sobre interfaces de rede elástica, consulte [Interfaces de rede elástica](#) no Guia EC2 do usuário da Amazon.

- Para obter mais informações sobre VPC endpoints, consulte Interface [VPC endpoints \(\) no Guia do usuário AWS PrivateLink da Amazon VPC](#).
- Para obter mais informações sobre operações de ElastiCache API, consulte [Operações de ElastiCache API](#).

Depois de criar uma interface VPC endpoint, se você habilitar nomes de host [DNS privados](#) para o endpoint, o endpoint padrão (<https://elasticache.Region.amazonaws.com>) e endpoint dualstack (<https://elasticache.Region.api.aws>) resolvem para seu VPC endpoint. Se você não habilitar nomes de host DNS privados, o Amazon VPC fornecerá um nome de endpoint DNS que poderá ser usado no seguinte formato:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Para obter mais informações, consulte [Interface VPC Endpoints \(AWS PrivateLink\)](#) no Guia do usuário da Amazon VPC. ElastiCache suporta fazer chamadas para todas as suas [ações de API](#) dentro da sua VPC.

Note

Os nomes de host DNS privados podem ser habilitados para apenas um endpoint da VPC na VPC. Se você quiser criar um endpoint da VPC adicional, o nome de host DNS privado deve ser desabilitado para esse endpoint.

Regiões do Privatelink disponíveis

Código	Local	Região
CPT	África (Cidade do Cabo)	AF-SOUTH-1
HKG	Ásia-Pacífico (Hong Kong)	AP-EAST-1
TPE	Ásia-Pacífico (Taipei)	AP-EAST-2

Código	Local	Região
NRT	Ásia-Pacífico (Tóquio)	AP-NORTHEAST-1
ICN	Ásia-Pacífico (Seul)	AP-NORTHEAST-2
KIX	Ásia-Pacífico (Osaka)	AP-NORTHEAST-3
BOM	Ásia-Pacífico (Mumbai)	AP-SOUTH-1
HYD	Ásia-Pacífico (Hyderabad)	AP-SOUTH-2
SIN	Ásia-Pacífico (Singapura)	AP-SOUTHEAST-1
SYD	Ásia-Pacífico (Sydney)	AP-SOUTHEAST-2
CGK	Ásia-Pacífico (Jacarta)	AP-SOUTHEAST-3
MEL	Ásia-Pacífico (Melbourne)	AP-SOUTHEAST-4
KUL	Ásia-Pacífico (Malásia)	AP-SOUTHEAST-5
BKK	Ásia-Pacífico (Tailândia)	AP-SOUTHEAST-7
YUL	Canadá (Central)	CA-CENTRAL-1
YYC	Oeste do Canadá (Calgary)	CA-WEST-1
BJS	China (Pequim)	CN-NORTH-1
ZHY	China (Ningxia)	CN-NORTHWEST-1
FRA	Europa (Frankfurt)	EU-CENTRAL-1
ZRH	Europa (Zurique)	EU-CENTRAL-2
ARN	Europa (Estocolmo)	EU-NORTH-1
MPX	Europa (Milão)	EU-SOUTH-1
ZAZ	Europa (Espanha)	EU-SOUTH-2

Código	Local	Região
DUB	Europa (Irlanda)	EU-WEST-1
LHR	Europa (Londres)	EU-WEST-2
CDG	Europa (Paris)	EU-WEST-3
TLV	Tel Aviv (Israel)	IL-CENTRAL-1
DXB	Oriente Médio (Emirados Árabes Unidos)	ME-CENTRAL-1
BAH	Oriente Médio (Barém)	ME-SOUTH-1
QRO	México (Central)	MX-CENTRAL-1
GRU	América do Sul (São Paulo)	SA-EAST-1
IAD	Leste dos EUA (Norte da Virgínia)	US-EAST-1
CMH	Leste dos EUA (Ohio)	US-EAST-2
OSU	AWS GovCloud (Leste dos EUA)	US-GOV-EAST-1
SFO	Oeste dos EUA (Norte da Califórnia)	US-WEST-1
PDX	Oeste dos EUA (Oregon)	US-WEST-2
PDT	AWS GovCloud (Oeste dos EUA)	US-WEST-1

Considerações sobre endpoints da VPC do

Antes de configurar uma interface VPC endpoint para endpoints de ElastiCache API da Amazon, certifique-se de revisar as [propriedades e limitações do endpoint da interface no Guia do usuário](#)

[da](#) Amazon VPC. Todas as operações de ElastiCache API relevantes para o gerenciamento de ElastiCache recursos da Amazon estão disponíveis em sua VPC usando AWS PrivateLink

As políticas de endpoint de VPC são compatíveis com endpoints de ElastiCache API. Por padrão, o acesso total às operações ElastiCache da API é permitido por meio do endpoint. Para mais informações, consulte [Controlar o acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

Criação de uma interface VPC endpoint para a API ElastiCache

Você pode criar um VPC endpoint para a ElastiCache API da Amazon usando o console da Amazon VPC ou o AWS CLI. Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário da Amazon VPC.

Depois de criar um endpoint da VPC de interface, será possível habilitar nomes de host DNS privados para o endpoint. Quando você fizer isso, o ElastiCache endpoint padrão da Amazon (<https://elasticache.Region.amazonaws.com>) e endpoint dualstack (<https://elasticache.Region.api.aws>) resolvem para seu VPC endpoint. Para as AWS regiões da China (Pequim) e China (Ningxia), você pode fazer solicitações de API com o VPC endpoint [elasticache.cn-north-1.amazonaws.com.cn](#) usando [elasticache.cn-north-1.api.amazonwebservices.com.cn](#) e para Pequim e e para Ningxia. [elasticache.cn-northwest-1.amazonaws.com.cn](#) [elasticache.cn-northwest-1.api.amazonwebservices.com.cn](#) Para mais informações, consulte [Acessar um serviço por um endpoint de interface](#) no Guia do usuário da Amazon VPC.

Criação de uma política de VPC endpoint para a API da Amazon ElastiCache

Você pode anexar uma política de endpoint ao seu VPC endpoint que controla o acesso à API. ElastiCache A política especifica o seguinte:

- A entidade principal que pode realizar ações.
- As ações que podem ser realizadas.
- Os recursos aos quais as ações podem ser aplicadas.

Para mais informações, consulte [Controlar o acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

Example Política de VPC endpoint para ações de ElastiCache API com Valkey ou Redis OSS

Veja a seguir um exemplo de uma política de endpoint para a ElastiCache API. Quando anexada a um endpoint, essa política concede acesso às ações de ElastiCache API listadas para todos os diretores em todos os recursos.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster",
      "elasticache:CreateSnapshot"
    ],
    "Resource": "*"
  }]
}
```

Example Política de VPC endpoint ElastiCache para ações da API Memcached

Veja a seguir um exemplo de uma política de endpoint para a ElastiCache API. Quando anexada a um endpoint, essa política concede acesso às ações de ElastiCache API listadas para todos os diretores em todos os recursos.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
  }]
}
```

Example Política de VPC endpoint que nega todo o acesso de uma conta especificada AWS

A política de VPC endpoint a seguir nega àAWS conta **123456789012** todo o acesso aos recursos que usam o endpoint. A política permite todas as ações de outras contas.

```
{
  "Statement": [{
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*",
    "Principal": "*"
  },
  {
    "Action": "*",
    "Effect": "Deny",
    "Resource": "*",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    }
  }
]
}
```

Sub-redes e grupos de sub-redes

Um grupo de sub-redes é um conjunto de sub-redes (normalmente privadas) que você pode designar para seus clusters baseados em nós em execução em um ambiente da nuvem privada virtual (VPC) da Amazon.

Se você criar um cluster baseado em nós em uma Amazon VPC, deverá usar um grupo de sub-redes. ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós.

ElastiCache fornece um grupo de IPv4 sub-redes padrão ou você pode optar por criar um novo. Para IPv6 isso, você precisa criar um grupo de sub-redes com um bloco IPv6 CIDR. Se você escolher pilha dupla, deverá selecionar um tipo de IP de descoberta, IPv6 ou IPv4.

ElastiCache O Serverless não usa um recurso de grupo de sub-redes e, em vez disso, obtém uma lista de sub-redes diretamente durante a criação.

Esta seção aborda como criar e aproveitar sub-redes e grupos de sub-redes para gerenciar o acesso aos seus recursos. ElastiCache

Para obter mais informações sobre o uso de grupos de sub-redes em um ambiente da Amazon VPC, consulte [Acessando seu ElastiCache cluster ou grupo de replicação](#).

Tópicos

- [Criação de um grupo de sub-redes](#)
- [Atribuição de um grupo de sub-redes a um cache](#)
- [Modificação de um grupo de sub-redes](#)
- [Exclusão de um grupo de sub-redes](#)

Criação de um grupo de sub-redes

Um grupo de sub-redes de cache é uma coleção de sub-redes que você pode querer designar aos seus caches em uma VPC. Ao executar um cache em uma VPC, é necessário selecionar um grupo de sub-redes de cache. Em seguida, o ElastiCache usa esse grupo de sub-rede de cache para atribuir endereços IP dentro dessa sub-rede a cada nó de cache do cache.

Quando você criar um novo grupo de sub-redes, observe o número de endereços IP disponíveis. Se a sub-rede tiver muito poucos endereços IP livres, talvez haja um limite no que diz respeito ao número de nós adicionais que é possível acrescentar a um cluster. Para resolver esse problema, você pode atribuir uma ou mais sub-redes a um grupo de sub-redes para ter um número suficiente de endereços IP na zona de disponibilidade do seu cluster. Depois disso, você pode adicionar mais nós ao seu cluster.

Se você escolher IPV4 como seu tipo de redes, um grupo de sub-rede padrão estará disponível ou é possível optar por criar um novo. O ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. Se você escolher pilha dupla ou IPV6, será direcionado para criar sub-redes de pilha dupla ou IPV6. Para obter mais informações sobre os tipos de rede, consulte [Escolhendo um tipo de rede em ElastiCache](#). Para obter mais informações, consulte [Criar uma sub-rede na sua VPC](#).

Os procedimentos a seguir mostram como criar um grupo de sub-rede chamado mysubnetgroup (console), a AWS CLI e a API do ElastiCache.

Criação de um grupo de sub-redes (console)

O procedimento a seguir mostra como criar um grupo de sub-redes (console).

Como criar um grupo de sub-redes (console)

1. Faça login no Console de Gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha Grupos de sub-redes.
3. Selecione Create subnet group (Criar grupo de sub-redes).
4. No assistente Criar grupo de sub-redes, faça o seguinte. Quando estiver satisfeito com as configurações, escolha Criar.
 - a. Na caixa Name (Nome), digite um nome para o seu grupo de sub-redes.
 - b. Na caixa Description (Descrição), digite uma descrição para seu grupo de sub-redes.

- c. Na caixa ID da VPC, escolha o Amazon VPC.
 - d. Todas as sub-redes são escolhidas por padrão. No painel Sub-redes selecionadas, clique em Gerenciar e selecione as zonas de disponibilidade ou as [Zonas locais](#) e os IDs das sub-redes privadas e selecione Escolher.
5. Na mensagem de confirmação exibida, escolha Close (Fechar).

Seu novo grupo de sub-rede aparece na lista Subnet Groups (Grupos de sub-redes) do console do ElastiCache. Na parte inferior da janela, você pode escolher o grupo de sub-redes para ver detalhes, como todas as sub-redes associadas a esse grupo.

Criação de um grupo de sub-redes (AWS CLI)

No prompt de comando, use o comando `create-cache-subnet-group` para criar um grupo de sub-redes.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Esse comando deve produzir um resultado semelhante ao seguinte:

```
{
  "CacheSubnetGroup": {
    "VpcId": "vpc-37c3cd17",
    "CacheSubnetGroupDescription": "Testing",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-53df9c3a",
        "SubnetAvailabilityZone": {
```

```
        "Name": "us-west-2a"
      }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
  }
}
```

Para obter mais informações, consulte o AWS CLI tópico [create-cache-subnet-group](#).

Atribuição de um grupo de sub-redes a um cache

Depois de criar um grupo de sub-redes, você pode iniciar um cache em um Amazon VPC. Para obter mais informações, consulte.

- Cluster do Memcached: para ativar um cluster de cache do Memcached, consulte [Criação de um cluster do Memcached \(console\)](#). Na etapa 7.a (Configurações avançadas do Memcached), escolha um grupo de sub-redes da VPC.
- Cluster Valkey ou Redis OSS autônomo: para ativar um cluster do Valkey ou Redis OSS de nó único, consulte [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#). Na etapa 7.a (Configurações avançadas do Redis OSS), escolha um grupo de sub-redes da VPC.
- Grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado): para ativar um grupo de replicação do Valkey ou Redis OSS (modo cluster desabilitado) em uma VPC, consulte [Criação de um grupo de replicação do Valkey ou Redis OSS \(modo cluster desabilitado\) do início](#). Na etapa 7.b (Configurações avançadas do Redis OSS), escolha um grupo de sub-redes da VPC.
- Grupo de replicação do Valkey ou Redis OSS (modo cluster habilitado): [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#). Na etapa 6.i (Configurações avançadas do Redis OSS), escolha um grupo de sub-redes da VPC.

Modificação de um grupo de sub-redes

Você pode modificar a descrição de um grupo de sub-redes ou modificar a lista de IDs de sub-rede associados ao grupo de sub-redes. Você não poderá excluir um ID de sub-rede de um grupo de sub-redes se um cache estiver usando essa sub-rede atualmente.

Os procedimentos a seguir mostram como modificar um grupo de sub-redes.

Modificação de grupos de sub-redes (console)

Para modificar um grupo de sub-redes

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Grupos de sub-redes.
3. Na lista de grupos de sub-redes, selecione o botão de opção daquele que você deseja modificar e escolha Modificar.
4. No painel Sub-redes selecionadas, escolha Gerenciar.
5. Faça qualquer alteração nas sub-redes selecionadas e clique em Escolher.
6. Clique em Salvar alterações para salvá-las.

Modificação de grupos de sub-redes (AWS CLI)

No prompt de comando, use o comando `modify-cache-subnet-group` para modificar um grupo de sub-redes.

Para 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"
```

Para Windows:

```
aws elasticache modify-cache-subnet-group ^
  --cache-subnet-group-name mysubnetgroup ^
  --cache-subnet-group-description "New description" ^
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```


Esse comando deve produzir um resultado semelhante ao seguinte:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte o AWS CLI tópico [modify-cache-subnet-group](#).

Exclusão de um grupo de sub-redes

Se você decidir que não precisa mais do seu grupo de sub-redes, poderá excluí-lo. Não será possível excluir um grupo de sub-redes se ele estiver sendo usado atualmente por um cache.

Os procedimentos a seguir mostram como excluir um grupo de sub-redes.

Exclusão de um grupo de sub-redes (console)

Para excluir um grupo de sub-redes

1. Faça login no Console de gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Grupos de sub-redes.
3. Na lista de grupos de sub-redes, escolha aquele que deseja excluir e selecione Delete.
4. Quando você precisar confirmar essa operação, digite o nome do grupo de sub-redes no campo de entrada de texto e escolha Excluir.

Exclusão de um grupo de sub-redes (AWS CLI)

Usando o AWS CLI, chame o comando `delete-cache-subnet-group` com o seguinte parâmetro:

- `--cache-subnet-group-name` *mysubnetgroup*

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup
```

Para Windows:

```
aws elasticache delete-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup
```

Este comando não produz saída.

Para obter mais informações, consulte o AWS CLI tópico [delete-cache-subnet-group](#).

Identity and Access Management para Amazon ElastiCache

AWS Identity and Access Management(IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar ElastiCache os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Gerenciar o acesso usando políticas](#)
- [Como a Amazon ElastiCache trabalha com o IAM](#)
- [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)
- [Solução de problemas de identidade e acesso do Amazon ElastiCache](#)
- [Controle de acesso](#)
- [Visão geral do gerenciamento de permissões de acesso aos seus ElastiCache recursos](#)

Público

A forma como você usa AWS Identity and Access Management(IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao administrador caso você não consiga acessar os recursos (consulte [Solução de problemas de identidade e acesso do Amazon ElastiCache](#)).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte [Como a Amazon ElastiCache trabalha com o IAM](#))
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#))

Autenticação com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM.Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM(IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Consulte mais informações sobre como fazer login em [Como fazer login em sua Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS.

Para acesso programático,AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Consulte mais informações em [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para obter mais informações, consulte [O que é o IAM Identity Center?](#) no Guia do usuário do Centro de Identidade do AWS IAM.

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte [Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Consulte mais informações em [Casos de uso de usuários do IAM](#) no Guia do usuário do IAM.

Perfis do IAM

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para obter mais informações, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

As funções do IAM são úteis para acesso de usuários federados, permissões temporárias de usuários do IAM, acesso entre contas, acesso entre serviços e aplicativos executados na Amazon. EC2 Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Consulte mais informações sobre documentos de política JSON em [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para obter mais informações, consulte [Limites de permissões para entidades do IAM](#) no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para obter mais informações, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations.
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para obter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

Como a Amazon ElastiCache trabalha com o IAM

Antes de usar o IAM para gerenciar o acesso ElastiCache, saiba com quais recursos do IAM estão disponíveis para uso ElastiCache.

Recursos do IAM que você pode usar com a Amazon ElastiCache

Recurso do IAM	ElastiCache apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Sim
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Sim

Para ter uma visão de alto nível de como ElastiCache e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWS os serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade para ElastiCache

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Saiba mais sobre todos os elementos que podem ser usados em uma política JSON consultando [Referência de Elementos de Política JSON do IAM](#) no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o ElastiCache

Para ver exemplos de políticas ElastiCache baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)

Políticas baseadas em recursos no ElastiCache

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Ações políticas para ElastiCache

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ElastiCache ações, consulte [Ações definidas pela Amazon ElastiCache](#) na Referência de autorização de serviço.

As ações de política ElastiCache usam o seguinte prefixo antes da ação:

```
elasticache
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
    "elasticache:action1",  
    "elasticache:action2"  
]
```

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "elasticache:Describe*"
```

Para ver exemplos de políticas ElastiCache baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)

Recursos políticos para ElastiCache

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de ElastiCache recursos e seus ARNs, consulte [Recursos definidos pela Amazon ElastiCache](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pela Amazon](#). ElastiCache

Para ver exemplos de políticas ElastiCache baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)

Chaves de condição de políticas para ElastiCache

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de ElastiCache condição, consulte [Chaves de condição da Amazon ElastiCache](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pela Amazon ElastiCache](#).

Para ver exemplos de políticas ElastiCache baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)

Listas de controle de acesso (ACLs) em ElastiCache

Suportes ACLs: Sim

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Controle de acesso baseado em atributos (ABAC) com ElastiCache

Compatível com ABAC (tags em políticas): sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte [Definir permissões com autorização do ABAC](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso por atributo \(ABAC\)](#) no Guia do usuário do IAM.

Usando credenciais temporárias com ElastiCache

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte [Credenciais de segurança temporárias no IAM](#) e [Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Permissões de entidade principal entre serviços para o ElastiCache

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

Funções de serviço para ElastiCache

Compatível com perfis de serviço: sim

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.

 Warning

Alterar as permissões de uma função de serviço pode interromper ElastiCache a funcionalidade. Edite perfis de serviço somente quando o ElastiCache fornecer orientação para fazê-lo.

Funções vinculadas a serviços para ElastiCache

Compatibilidade com perfis vinculados a serviços: sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para a Amazon ElastiCache

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do ElastiCache. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte [Criar políticas do IAM \(console\)](#) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos por ElastiCache, incluindo o formato de cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição para a Amazon ElastiCache](#) na Referência de autorização de serviço. ARNs

Tópicos

- [Práticas recomendadas de política](#)
- [Usando o ElastiCache console](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir ElastiCache recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.

- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o ElastiCache console

Para acessar o ElastiCache console da Amazon, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os ElastiCache recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o ElastiCache console, anexe também a política ElastiCache ConsoleAccess ou a política ReadOnlyAWS gerenciada às entidades. Para obter informações, consulte [Adicionar permissões a um usuário](#) no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
}
]
```

Solução de problemas de identidade e acesso do Amazon ElastiCache

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o ElastiCache e o IAM.

Tópicos

- [Não tenho autorização para executar uma ação no ElastiCache](#)
- [Não estou autorizado a executar iam:PassRole](#)
- [Desejo permitir que pessoas fora da minha conta da AWS acessem meus recursos do ElastiCache](#)

Não tenho autorização para executar uma ação no ElastiCache

Se o Console de gerenciamento da AWS informar que você não foi autorizado a executar uma ação, você deve entrar em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu o seu nome de usuário e senha.

O erro do exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um recurso do *my-example-widget* fictício, mas não tem as permissões fictícias do elasticsearch: *GetWidget*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticache:GetWidget on resource: my-example-widget
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *my-example-widget* usando a ação elasticsearch: *GetWidget*.

Não estou autorizado a executar iam:PassRole

Se você receber um erro informando que não tem autorização para executar a ação `iam:PassRole`, as políticas deverão ser atualizadas para permitir que você passe um perfil para o ElastiCache.

Alguns Serviços da AWS permitem que você passe um perfil existente para o serviço, em vez de criar um perfil de serviço ou perfil vinculado ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no ElastiCache. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Desejo permitir que pessoas fora da minha conta da AWS acessem meus recursos do ElastiCache

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o ElastiCache oferece suporte a esses recursos, consulte [Como a Amazon ElastiCache trabalha com o IAM](#).
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte [Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você](#) no Guia de usuário do IAM.
- Para saber como conceder acesso a seus recursos para Contas da AWS de terceiros, consulte [Fornecimento de acesso a Contas da AWS pertencentes a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Controle de acesso

Você pode ter credenciais válidas para autenticar suas solicitações, mas, a menos que tenha permissões, não poderá criar ou acessar ElastiCache recursos. Por exemplo, você deve ter permissões para criar um ElastiCache cluster.

As seções a seguir descrevem como gerenciar permissões para ElastiCache. Recomendamos que você leia a visão geral primeiro.

- [Visão geral do gerenciamento de permissões de acesso aos seus ElastiCache recursos](#)
- [Usando políticas baseadas em identidade \(políticas do IAM\) para a Amazon ElastiCache](#)

Visão geral do gerenciamento de permissões de acesso aos seus ElastiCache recursos

Cada AWS recurso pertence a uma AWS conta, e as permissões para criar ou acessar um recurso são regidas por políticas de permissões. Um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções). Além disso, a Amazon ElastiCache também oferece suporte para anexar políticas de permissões aos recursos.

Note

Um administrador da conta (ou usuário administrador) é um usuário com privilégios de administrador. Para obter mais informações, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em Centro de Identidade do AWS IAM:

Crie um conjunto de permissões. Siga as instruções em [Criação de um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM.

- Usuários gerenciados no IAM com provedor de identidades:

Crie um perfil para a federação de identidades. Siga as instruções em [Criando um perfil para um provedor de identidades de terceiros \(federação\)](#) no Guia do Usuário do IAM.

- Usuários do IAM:

- Crie um perfil que seu usuário possa assumir. Siga as instruções em [Criação de um perfil para um usuário do IAM](#) no Guia do usuário do IAM.
- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em [Adição de permissões a um usuário \(console\)](#) no Guia do usuário do IAM.

Tópicos

- [ElastiCache Recursos e operações da Amazon](#)
- [Informações sobre propriedade de recursos](#)
- [Gerenciar acesso aos recursos da](#)

- [AWS políticas gerenciadas para a Amazon ElastiCache](#)
- [Usando políticas baseadas em identidade \(políticas do IAM\) para a Amazon ElastiCache](#)
- [Permissões em nível de recurso](#)
- [Uso de chaves de condição](#)
- [Uso de funções vinculadas ao serviço para o Amazon ElastiCache](#)
- [Permissões da API do ElastiCache: referência de ações, recursos e condições](#)

ElastiCache Recursos e operações da Amazon

Para ver uma lista dos tipos de ElastiCache recursos e seus ARNs, consulte [Recursos definidos pela Amazon ElastiCache](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pela Amazon](#). ElastiCache

Informações sobre propriedade de recursos

O proprietário do recurso é a AWS conta que criou o recurso. Ou seja, o proprietário do recurso é a AWS conta da entidade principal que autentica a solicitação que cria o recurso. Uma entidade principal pode ser a conta raiz, um usuário do IAM ou uma função do IAM. Os seguintes exemplos mostram como isso funciona:

- Suponha que você use as credenciais da conta raiz da sua AWS conta para criar um cluster de cache. Nesse caso, sua AWS conta é a proprietária do recurso. Em ElastiCache, o recurso é o cluster.
- Suponha que você crie um usuário do IAM em sua AWS conta e conceda permissões para criar um cluster para esse usuário. Nesse caso, o usuário pode criar um cluster. No entanto, sua AWS conta, à qual o usuário pertence, é proprietária do recurso de cluster.
- Suponha que você crie uma função do IAM em sua AWS conta com permissões para criar um cluster. Nesse caso, qualquer pessoa que possa assumir a função poderá criar um cluster. Sua AWS conta, à qual a função pertence, é proprietária do recurso de cluster de cache.

Gerenciar acesso aos recursos da

Uma política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções disponíveis para a criação de políticas de permissões.

 Note

Esta seção discute o uso do IAM no contexto da Amazon ElastiCache. Não são fornecidas informações detalhadas sobre o serviço IAM. Para obter a documentação completa do IAM, consulte [O que é o IAM?](#) no Guia do usuário do IAM. Para obter mais informações sobre a sintaxe e as descrições da política do IAM, consulte a [Referência de políticas do AWS IAM](#) no Guia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas como políticas baseadas em identidade (políticas do IAM). As políticas anexadas a um recurso são chamadas de políticas baseadas em recursos.

Tópicos

- [Políticas baseadas em identidade \(políticas do IAM\)](#)
- [Especificar elementos da política: ações, efeitos, recursos e entidades principais](#)
- [Especificar condições em uma política](#)

Políticas baseadas em identidade (políticas do IAM)

É possível anexar políticas a identidades do IAM. Por exemplo, você pode fazer o seguinte:

- Anexar uma política de permissões a um usuário ou grupo na sua conta: um administrador de conta pode usar uma política de permissões associada a determinado usuário para conceder permissões. Nesse caso, as permissões são para que esse usuário crie um ElastiCache recurso, como um cluster, um grupo de parâmetros ou um grupo de segurança.
- Anexar uma política de permissões a uma função: você pode anexar uma política de permissões baseada em identidade a um perfil do IAM para conceder permissões entre contas. Por exemplo, o administrador na Conta A pode criar uma função para conceder permissões entre contas a outra AWS conta (por exemplo, Conta B) ou a um AWS serviço da seguinte forma:
 1. Um administrador da Conta A cria uma função do IAM e anexa uma política de permissões à função que concede permissões em recursos da Conta A.
 2. Um administrador da Conta A anexa uma política de confiança à função identificando a Conta B como a entidade principal, que pode assumir a função.
 3. O administrador da Conta B pode então delegar permissões para assumir a função a qualquer usuário na Conta B. Isso permite que os usuários da Conta B criem ou acessem recursos na

Conta A. Em alguns casos, talvez você queira conceder permissões a um AWS serviço para assumir a função. Para oferecer suporte a essa abordagem, o principal da política de confiança também pode ser um principal de serviço da AWS.

Para obter mais informações sobre o uso do IAM para delegar permissões, consulte [Gerenciamento de acesso](#) no Guia do usuário do IAM.

Veja a seguir um exemplo de política que permite que um usuário execute a `DescribeCacheClusters` ação AWS em sua conta. ElastiCache também suporta a identificação de recursos específicos usando o recurso ARNs para ações de API. Essa abordagem também é chamada de permissões no nível do recurso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribeCacheClusters",
      "Effect": "Allow",
      "Action": [
        "elasticache:DescribeCacheClusters"
      ],
      "Resource": "arn:aws:iam::*:role/*"
    }
  ]
}
```

Para obter mais informações sobre o uso de políticas baseadas em identidade com ElastiCache, consulte [Usando políticas baseadas em identidade \(políticas do IAM\) para a Amazon ElastiCache](#). Para obter mais informações sobre usuários, grupos, funções e permissões, consulte [Identidades \(usuários, grupos e funções\)](#) no Guia do usuário do IAM.

Especificar elementos da política: ações, efeitos, recursos e entidades principais

Para cada ElastiCache recurso da Amazon (consulte [ElastiCache Recursos e operações da Amazon](#)), o serviço define um conjunto de operações de API (consulte [Ações](#)). Para conceder permissões para essas operações de API, ElastiCache defina um conjunto de ações que você pode especificar em uma política. Por exemplo, para o recurso de ElastiCache cluster, as seguintes ações são definidas:

CreateCacheClusterDeleteCacheCluster, DescribeCacheCluster e. A execução de uma operação de API pode exigir permissões para mais de uma ação.

Estes são os elementos de política mais básicos:

- **Recurso:** em uma política, você usa um Amazon Resource Name (ARN – Nome do recurso da Amazon) para identificar o recurso a que a política se aplica. Para obter mais informações, consulte [ElastiCache Recursos e operações da Amazon](#).
- **Ação:** você usa palavras-chave de ação para identificar operações de recursos que deseja permitir ou negar. Por exemplo, dependendo do especificadoEffect, a elasticache:CreateCacheCluster permissão permite ou nega ao usuário permissões para realizar a ElastiCache CreateCacheCluster operação da Amazon.
- **Efeito:** você especifica o efeito quando o usuário solicita a ação específica, que pode ser permitir ou negar. Se você não conceder (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar acesso explicitamente a um recurso. Por exemplo, você poderia fazer isso para garantir que um usuário não possa acessar o recurso, mesmo se uma política diferente conceder o acesso.
- **Entidade principal:** em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é a entidade principal implícita. Para as políticas baseadas em recursos, você especifica quais usuários, contas, serviços ou outras entidades deseja que recebam permissões (isso se aplica somente a políticas baseadas em recursos).

Para saber mais sobre a sintaxe e as descrições de políticas do IAM, consulte a [Referência de políticas do AWS IAM da](#) no Guia do usuário do IAM.

Para ver uma tabela mostrando todas as ações de ElastiCache API da Amazon, consulte [Permissões da API do ElastiCache: referência de ações, recursos e condições](#).

Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da política do IAM para especificar as condições de quando uma política deverá entrar em vigor. Por exemplo, é recomendável aplicar uma política somente após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte [Condition](#) no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Para usar chaves ElastiCache de condição específicas, consulte [Uso de chaves de condição](#). Existem chaves AWS de condição

abrangentes que você pode usar conforme apropriado. Para obter uma lista completa AWS de chaves abrangentes, consulte [Chaves disponíveis para condições](#) no Guia do usuário do IAM.

AWS políticas gerenciadas para a Amazon ElastiCache

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. As políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as [políticas gerenciadas pelo cliente](#) que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. É mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para mais informações, consulte [Políticas gerenciadas pela AWS](#) no Guia do usuário do IAM.

AWS política gerenciada: ElastiCacheServiceRolePolicy

Você não pode se vincular ElastiCacheServiceRolePolicy às suas entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que permite ElastiCache realizar ações em seu nome.

Essa política permite ElastiCache gerenciar AWS recursos em seu nome conforme necessário para gerenciar seu cache:

- `ec2`— Gerencie recursos EC2 de rede para se conectar aos nós de cache, incluindo endpoints VPC (para caches sem servidor), interfaces de rede elásticas () (para clusters baseados em nós ENIs) e grupos de segurança.
- `cloudwatch`— Emitir dados métricos do serviço para o CloudWatch
- `outposts`— Permitir a criação de nós de cache em AWS Outposts.

Você pode encontrar a [ElastiCacheServiceRolePolicy](#) política no console do IAM e [ElastiCacheServiceRolePolicy](#) no Guia de referência de políticas AWS gerenciadas.

AWS política gerenciada: AmazonElastiCacheFullAccess

É possível anexar a política AmazonElastiCacheFullAccess às suas identidades do IAM.

Essa política permite que os diretores tenham acesso total ao ElastiCache uso do AWS Management Console:

- `elasticache`— Acesse tudo APIs.
- `iam`: crie uma função vinculada ao serviço necessária para a operação de serviço.
- `ec2`— Descreva os EC2 recursos dependentes necessários para a criação de cache (VPC, sub-rede, grupo de segurança) e permita a criação de endpoints de VPC (para caches sem servidor).
- `kms`— Permitir o uso do formulário gerenciado pelo cliente CMKs . `encryption-at-rest`
- `cloudwatch`— Permita o acesso às ElastiCache métricas para exibir métricas no console.
- `application-autoscaling`: permita o acesso para descrever as políticas de escalonamento automático para caches.
- `logs`: usado para preencher fluxos de log para a funcionalidade de entrega de logs no console.
- `firehose`: usado para preencher fluxos de log de entrega para a funcionalidade de entrega de logs no console.
- `s3`: usado para preencher buckets S3 para a funcionalidade de restauração de instantâneos no console.
- `outposts`— Usado para preencher AWS Outposts para criação de cache no console.
- `sns`: usado para preencher tópicos do SNS para funcionalidade de notificação no console.

Você pode encontrar a [AmazonElastiCacheFullAccess](#) política no console do IAM e [AmazonElastiCacheFullAccess](#) no Guia de referência de políticas AWS gerenciadas.

AWS política gerenciada: AmazonElastiCacheReadOnlyAccess

É possível anexar a política AmazonElastiCacheReadOnlyAccess às suas identidades do IAM.

Essa política permite que os diretores tenham acesso somente para leitura ao ElastiCache uso do AWS Management Console:

- `elasticache`— Acesso somente para leitura `Describe` APIs.

Você pode encontrar a [AmazonElastiCacheReadOnlyAccess](#) política no console do IAM e [AmazonElastiCacheReadOnlyAccess](#) no Guia de referência de políticas AWS gerenciadas.

ElastiCache atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas ElastiCache desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página [Histórico do ElastiCache](#) documento.

Alteração	Descrição	Data
AmazonElastiCacheFullAccess : atualizar para uma política existente	ElastiCache adicionou novas permissões para permitir o dimensionamento vertical para a ação <code>elasticache:ModifyCacheCluster</code> . <code>MemCached</code>	27 de março de 2025
AmazonElastiCacheFullAccess – atualização para uma política existente	ElastiCache adicionou novas permissões para permitir o gerenciamento de caches sem servidor e para permitir o uso de todos os recursos do serviço por meio do console.	27 de novembro de 2023
ElastiCacheServiceRolePolicy – atualização para uma política existente	ElastiCache adicionou novas permissões para permitir o gerenciamento de VPC endpoints para recursos de cache sem servidor.	27 de novembro de 2023
ElastiCache começou a rastrear alterações	ElastiCache começou a rastrear as mudanças em suas políticas AWS gerenciadas.	7 de fevereiro de 2020

Usando políticas baseadas em identidade (políticas do IAM) para a Amazon ElastiCache

Este tópico fornece exemplos de políticas baseadas em identidade em que um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções).

Important

Recomendamos que você leia primeiro os tópicos que explicam os conceitos básicos e as opções para gerenciar o acesso aos ElastiCache recursos da Amazon. Para obter mais informações, consulte [Visão geral do gerenciamento de permissões de acesso aos seus ElastiCache recursos](#).

As seções neste tópico abrangem o seguinte:

- [AWS políticas gerenciadas para a Amazon ElastiCache](#)
- [Exemplos de política gerenciada pelo cliente](#)

A seguir é mostrado um exemplo de política de permissões ao usar o Redis OSS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowClusterPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache",
        "elasticache:CreateCacheCluster",
        "elasticache:DescribeServerlessCaches",
        "elasticache:DescribeReplicationGroups",
        "elasticache:DescribeCacheClusters",
        "elasticache:ModifyServerlessCache",
        "elasticache:ModifyReplicationGroup",
        "elasticache:ModifyCacheCluster"
      ]
    }
  ]
}
```

```

        ],
        "Resource": "*"
    },
    {
        "Sid": "AllowUserToPassRole",
        "Effect": "Allow",
        "Action": [ "iam:PassRole" ],
        "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
    }
]
}

```

A seguir, é mostrado um exemplo de política de permissões ao usar o Memcached.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowClusterPermissions",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache",
            "elasticache:CreateCacheCluster",
            "elasticache:DescribeServerlessCaches",
            "elasticache:DescribeCacheClusters",
            "elasticache:ModifyServerlessCache",
            "elasticache:ModifyCacheCluster"
        ],
        "Resource": "*"
    },
    {
        "Sid": "AllowUserToPassRole",
        "Effect": "Allow",
        "Action": [ "iam:PassRole" ],
        "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
    }
    ]
}

```

A política tem duas instruções:

- A primeira declaração concede permissões para as ElastiCache ações da Amazon (`elasticache:Create*`, `elasticache:Describe*`, `elasticache:Modify*`)
- A segunda instrução concede permissões para a ação do IAM (`iam:PassRole`) no nome da função do IAM especificado no final do valor `Resource`.

A política não especifica o elemento `Principal` porque, em uma política baseada em identidade, a entidade principal que obtém as permissões não é especificada. Quando você anexar uma política um usuário, o usuário será a entidade principal implícita. Quando você anexa uma política de permissões a um perfil do IAM, a entidade principal identificada na política de confiança do perfil obtém as permissões.

Para ver uma tabela mostrando todas as ações de ElastiCache API da Amazon e os recursos aos quais elas se aplicam, consulte [Permissões da API do ElastiCache: referência de ações, recursos e condições](#).

Exemplos de política gerenciada pelo cliente

Se você não estiver usando uma política padrão e optar por usar uma política gerenciada personalizada, realize uma destas ações: Você deve ter permissões para chamar `iam:createServiceLinkedRole` (para obter mais informações, consulte [Exemplo 4: permitir que um usuário chame a CreateServiceLinkedRole API IAM](#)). Ou você deveria ter criado uma função ElastiCache vinculada ao serviço.

Quando combinadas com as permissões mínimas necessárias para usar o ElastiCache console da Amazon, as políticas de exemplo nesta seção concedem permissões adicionais. Os exemplos também são relevantes para o AWS SDKs e AWS CLI o.

Para obter instruções sobre como configurar usuários e grupos do IAM, consulte [Criação do seu primeiro usuário do IAM e grupo de administradores](#) no Guia do usuário do IAM.

Important

Sempre teste suas políticas do IAM completamente antes de usá-las em produção. Algumas ElastiCache ações que parecem simples podem exigir outras ações para apoiá-las quando você estiver usando o ElastiCache console. Por exemplo, `elasticache:CreateCacheCluster` concede permissões para criar ElastiCache

clusters. No entanto, para realizar essa operação, o ElastiCache console usa várias List ações Describe para preencher as listas do console.

Exemplos

- [Exemplo 1: permitir que um usuário tenha acesso somente de leitura aos recursos ElastiCache](#)
- [Exemplo 2: Permitir que um usuário execute tarefas comuns de administrador do ElastiCache sistema](#)
- [Exemplo 3: permitir que um usuário acesse todas as ações ElastiCache da API](#)
- [Exemplo 4: permitir que um usuário chame a CreateServiceLinkedRole API IAM](#)
- [Exemplo 5: permitir que um usuário se conecte ao cache sem servidor usando a autenticação do IAM](#)

Exemplo 1: permitir que um usuário tenha acesso somente de leitura aos recursos ElastiCache

A política a seguir concede ElastiCache ações de permissões que permitem ao usuário listar recursos. Normalmente, você anexa esse tipo de política de permissões a um grupo de gerentes.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECReadOnly",
    "Effect": "Allow",
    "Action": [
      "elasticache:Describe*",
      "elasticache:List*"
    ],
    "Resource": "*"
  }]
}
```

Exemplo 2: Permitir que um usuário execute tarefas comuns de administrador do ElastiCache sistema

Entre as tarefas do administrador do sistema comuns estão a modificação de recursos. Um administrador do sistema também pode querer obter informações sobre eventos do ElastiCache . A política a seguir concede permissões ao usuário para realizar ElastiCache ações para essas tarefas comuns do administrador do sistema. Normalmente, você anexa esse tipo de política de permissões ao grupo de administradores do sistema.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowMutations",
    "Effect": "Allow",
    "Action": [
      "elasticache:Modify*",
      "elasticache:Describe*",
      "elasticache:ResetCacheParameterGroup"
    ],
    "Resource": "*"
  }]
}
```

Exemplo 3: permitir que um usuário acesse todas as ações ElastiCache da API

A política a seguir permite que um usuário acesse todas as ElastiCache ações. Recomendamos que você conceda esse tipo de política de permissões apenas a um usuário administrador.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowAll",
    "Effect": "Allow",
    "Action": [
```

```
        "elasticache:*"  
    ],  
    "Resource": "*"  
}  
]  
}
```

Exemplo 4: permitir que um usuário chame a CreateServiceLinkedRole API IAM

A política a seguir permite que o usuário chame a API CreateServiceLinkedRole do IAM. Recomendamos que você conceda esse tipo de política de permissões ao usuário que invoca operações ElastiCache mutativas.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "CreateSLRAllows",  
      "Effect": "Allow",  
      "Action": [  
        "iam:CreateServiceLinkedRole"  
      ],  
      "Resource": "*",  
      "Condition": {  
        "StringLike": {  
          "iam:AWSServiceName": "elasticache.amazonaws.com"  
        }  
      }  
    }  
  ]  
}
```

Exemplo 5: permitir que um usuário se conecte ao cache sem servidor usando a autenticação do IAM

A política a seguir permite que qualquer usuário se conecte a qualquer cache sem servidor usando a autenticação do IAM entre 2023-04-01 e 2023-06-30.

JSON

```
{
  "Version": "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
      ],
      "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},
        "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
      }
    },
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:user:*"
      ]
    }
  ]
}
```

Permissões em nível de recurso

Você pode restringir o escopo das permissões especificando recursos em uma política do IAM. Muitas ações de API do ElastiCache oferecem suporte a um tipo de recurso que varia de acordo com o comportamento da ação. Cada instrução de política do IAM concede permissão a uma ação realizada em um recurso. Quando a ação não atua em um recurso indicado, ou quando você concede permissão para executar a ação em todos os recursos, o valor do recurso na política é um curinga (*). Para muitas ações de API, restrinja os recursos que um usuário pode modificar especificando o Amazon Resource Name (ARN – Nome de recurso da Amazon) de um recurso ou um padrão de ARN correspondente a vários recursos. Para restringir as permissões por recurso especifique o recurso por ARN.

Para ver uma lista dos tipos de recursos do ElastiCache e seus ARNs, consulte [Resources Defined by Amazon ElastiCache](#) na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Actions Defined by Amazon ElastiCache](#).

Exemplos

- [Exemplo 1: permitir que um usuário tenha acesso total a tipos de recurso específicos do ElastiCache](#)
- [Exemplo 2: negue a um usuário acesso a um cache sem servidor.](#)

Exemplo 1: permitir que um usuário tenha acesso total a tipos de recurso específicos do ElastiCache

A seguinte política permite explicitamente todos os recursos do tipo cache sem servidor.

```
{
  "Sid": "Example1",
  "Effect": "Allow",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
  ]
}
```

Exemplo 2: negue a um usuário acesso a um cache sem servidor.

O exemplo a seguir nega explicitamente o acesso a um determinado cache sem servidor.

```
{
  "Sid": "Example2",
  "Effect": "Deny",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
  ]
}
```

Uso de chaves de condição

Você pode especificar as condições que determinam como uma política do IAM entra em vigor. Em ElastiCache, você pode usar o `Condition` elemento de uma política JSON para comparar chaves

no contexto da solicitação com valores de chave que você especifica em sua política. Para obter mais informações, consulte [Elementos da política JSON do IAM: condição](#).

Para ver uma lista de chaves de ElastiCache condição, consulte [Chaves de condição da Amazon ElastiCache](#) na Referência de autorização de serviço.

Para obter uma lista de todas as chaves de condição globais, consulte [Chaves de contexto de condição global da AWS](#).

Usando ElastiCache com chaves de condição AWS globais

Ao usar [chaves de condição AWS globais](#) que exigem ElastiCache o [Principal](#), use uma OR condição com os dois Principais: `elasticache.amazonaws.com` e `ec.amazonaws.com`

Note

Se você não adicionar os dois Diretores ElastiCache, a ação pretendida de “Permitir” ou “Negar” não será aplicada corretamente para nenhum recurso listado em sua política.

Exemplo de política com a chave de condição global `aws:CalledVia`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:CalledVia": [
            "ec.amazonaws.com",
            "elasticache.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

```
}
```

Especificação de condições: uso de chaves de condição

Para implementar um controle refinado, grave uma política de permissões do IAM que especifique condições para controlar um conjunto de parâmetros individuais em determinadas solicitações. Em seguida, aplique a política aos usuários, grupos ou funções do IAM que você criar usando o console do IAM.

Para aplicar uma condição, adicione as informações da condição à declaração de política do IAM. No exemplo a seguir, você especifica a condição de que qualquer cluster baseado em nós criado será do tipo de nó cache `.r5.large`.

Note

- Para criar elementos `Condition` usando chaves de condição do tipo `String`, use os operadores de condição que não diferenciam maiúsculas de minúsculas `StringEqualsIgnoreCase` ou `StringNotEqualsIgnoreCase` para comparar uma chave com um valor de string.
- ElastiCache processa os argumentos de entrada para `CacheNodeType` e sem `CacheParameterGroupName` distinção entre maiúsculas e minúsculas. Por esse motivo, os operadores de condição de string `StringEqualsIgnoreCase` e `StringNotEqualsIgnoreCase` devem ser usados nas políticas de permissões que fazem referência a eles.

O exemplo a seguir mostra um exemplo dessa política de permissões ao usar o Valkey ou o Redis OSS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
```

```

        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:CacheNodeType": [
                "cache.r5.large"
            ]
        }
    }
}
]
}

```

A seguir, mostramos um exemplo dessa política de permissões ao usar o Memcached.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",

```

```

        "arn:aws:elasticache:*:*:subnetgroup:*"
    ],
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateCacheCluster"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:cluster:*"
        ],
        "Condition": {
            "StringEquals": {
                "elasticache:CacheNodeType": [
                    "cache.r5.large"
                ]
            }
        }
    }
]
}

```

Para obter mais informações, consulte [Marcando seus recursos ElastiCache](#).

Para obter mais informações sobre a utilização de operadores de condição de política, consulte [Permissões da API do ElastiCache: referência de ações, recursos e condições](#).

Políticas de exemplo: uso de condições para controle de acesso refinado

Esta seção mostra exemplos de políticas para implementar um controle de acesso refinado nos parâmetros listados anteriormente. ElastiCache

1. `elasticacheMaximumDataStorage`: especifique o armazenamento máximo de dados de um cache sem servidor. Usando as condições fornecidas, o cliente não pode criar caches que possam armazenar mais do que uma quantidade específica de dados.

JSON

```

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

```

```

        "Sid": "AllowDependentResources",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
            "arn:aws:elasticache:*:*:snapshot:*",
            "arn:aws:elasticache:*:*:usergroup:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:serverlesscache:*"
        ],
        "Condition": {
            "NumericLessThanEquals": {
                "elasticache:MaximumDataStorage": "30"
            },
            "StringEquals": {
                "elasticache:DataStorageUnit": "GB"
            }
        }
    }
]
}

```

2. `elasticache:Maximum ECPUPer Second`: especifique o valor máximo de ECPU por segundo de um cache sem servidor. Usando as condições fornecidas, o cliente não pode criar caches que possam executar mais do que um número específico de ECPUs por segundo.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [

```

```

        "elasticache:CreateServerlessCache"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateServerlessCache"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
        "NumericLessThanEquals": {
            "elasticache:MaximumECPUperSecond": "100000"
        }
    }
}
]
}

```

3. `elasticacheCacheNodeType`: Especifique quais `NodeType` (is) um usuário pode criar. Usando as condições fornecidas, o cliente pode especificar um valor único ou um intervalo para um tipo de nó.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        }
    ]
}

```



```

    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.t2.micro",
            "cache.t2.medium"
          ]
        }
      }
    }
  ]
}

```

4. `elasticache:CacheNodeType`: Com o Memcached, especifique quais `NodeType` (is) um usuário pode criar. Usando as condições fornecidas, o cliente pode especificar um valor único ou um intervalo para um tipo de nó.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    }
  ],
}

```

```

    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.t2.micro",
            "cache.t2.medium"
          ]
        }
      }
    }
  ]
}

```

5. `elasticacheNumNodeGroups`: Crie um grupo de replicação com menos de 20 grupos de nós.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [

```

```

        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "NumericLessThanEquals": {
            "elasticache:NumNodeGroups": "20"
        }
    }
}
]
}

```

6. `elasticacheReplicasPerNodeGroup`: especifique as réplicas por nó entre 5 e 10.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "NumericGreaterThanEquals": {
                    "elasticache:ReplicasPerNodeGroup": "5"
                },
                "NumericLessThanEquals": {
                    "elasticache:ReplicasPerNodeGroup": "10"
                }
            }
        }
    ]
}

```

```

    }
  }
]
}

```

7. `elasticache:EngineVersion`: Especifique o uso da versão 5.0.6 do mecanismo.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineVersion": "5.0.6"
        }
      }
    }
  ]
}

```

8. `elasticacheEngineVersion`: Especifique o uso do mecanismo Memcached versão 1.6.6

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineVersion": "1.6.6"
        }
      }
    }
  ]
}
```

9. `elasticacheEngineType`: especifique usando somente um mecanismo OSS Valkey ou Redis.

JSON

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

```

        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:EngineType": "redis"
        }
    }
}
]
}

```

10. `elasticacheAtRestEncryptionEnabled`: especifique que os grupos de replicação seriam criados somente com a criptografia ativada.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",

```

```

        "arn:aws:elasticache:*:*:subnetgroup:*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "Bool": {
            "elasticache:AtRestEncryptionEnabled": "true"
        }
    }
}
]
}

```

11 dor elástica: TransitEncryptionEnabled

- Defina a chave de `elasticache:TransitEncryptionEnabled` condição `false` para a [CreateReplicationGroup](#) ação para especificar que grupos de replicação só podem ser criados quando o TLS não está sendo usado:

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {

```

```

    "Effect": "Allow",
    "Action": [
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "Bool": {
            "elasticache:TransitEncryptionEnabled": "false"
        }
    }
}

```

Quando a chave de `elasticache:TransitEncryptionEnabled` condição é definida `false` em uma política para a [CreateReplicationGroup](#) ação, uma `CreateReplicationGroup` solicitação só será permitida se o TLS não estiver sendo usado (ou seja, se a solicitação não incluir um `TransitEncryptionEnabled` parâmetro definido como `true` ou um `TransitEncryptionMode` parâmetro definido como `required`).

- b. Defina a chave de `elasticache:TransitEncryptionEnabled` condição `true` para que a [CreateReplicationGroup](#) ação especifique que grupos de replicação só podem ser criados quando o TLS está sendo usado:

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {

```



```

    "Effect": "Allow",
    "Action": [
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "Bool": {
            "elasticache:TransitEncryptionEnabled": "true"
        }
    }
}

```

Quando a chave de `elasticache:TransitEncryptionEnabled` condição é definida `true` em uma política para a [CreateReplicationGroup](#) ação, uma `CreateReplicationGroup` solicitação só será permitida se a solicitação incluir um `TransitEncryptionEnabled` parâmetro definido como `true` e um `TransitEncryptionMode` parâmetro definido como `required`.

- c. Defina `elasticache:TransitEncryptionEnabled` como `true` para a ação `ModifyReplicationGroup` a fim de especificar que os grupos de replicação só podem ser modificados quando o TLS está sendo usado:

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:ModifyReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "BoolIfExists": {
                    "elasticache:TransitEncryptionEnabled": "true"
                }
            }
        }
    ]
}

```

```

    }
  }
]
}

```

Quando a chave de `elasticache:TransitEncryptionEnabled` condição é definida `true` em uma política para a [ModifyReplicationGroup](#) ação, uma `ModifyReplicationGroup` solicitação só será permitida se a solicitação incluir um `TransitEncryptionMode` parâmetro definido como `required`. Opcionalmente, o parâmetro `TransitEncryptionEnabled` definido como `true` também pode ser incluído, mas não é necessário nesse caso para habilitar o TLS.

`elasticacheAutomaticFailoverEnabled`: especifique que os grupos de replicação seriam criados somente com o failover automático ativado.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:AutomaticFailoverEnabled": "true"
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

13 `elasticache:multi AZEnabled`: especifique que grupos de replicação não podem ser criados com o Multi-AZ desativado.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:MultiAZEnabled": "false"
        }
      }
    }
  ]
}

```

14. `elasticacheClusterModeEnabled`: especifique que grupos de replicação só podem ser criados com o modo de cluster ativado.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:ClusterModeEnabled": "true"
        }
      }
    }
  ]
}
```

15. `elasticacheAuthTokenEnabled`: especifique que grupos de replicação só podem ser criados com o token AUTH ativado.

JSON

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "elasticache:CreateCacheCluster",
          "elasticache:CreateReplicationGroup"
        ],
        "Resource": [
          "arn:aws:elasticache:*:*:parametergroup:*",
          "arn:aws:elasticache:*:*:subnetgroup:*"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "elasticache:CreateCacheCluster",
          "elasticache:CreateReplicationGroup"
        ],
        "Resource": [
          "arn:aws:elasticache:*:*:cluster:*",
          "arn:aws:elasticache:*:*:replicationgroup:*"
        ],
        "Condition": {
          "Bool": {
            "elasticache:AuthTokenEnabled": "true"
          }
        }
      }
    ]
  }
}

```

elasticacheSnapshotRetentionLimit: especifique o número de dias (ou mínimo/máximo) para manter o instantâneo. A política abaixo impõe o armazenamento de backups por pelo menos 30 dias.

JSON

```

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

```

```

        "Effect": "Allow",
        "Action": [
            "elasticache:CreateCacheCluster",
            "elasticache:CreateReplicationGroup"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:parametergroup:*",
            "arn:aws:elasticache:*:*:subnetgroup:*"
        ]
    },

    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateCacheCluster",
            "elasticache:CreateReplicationGroup",
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:cluster:*",
            "arn:aws:elasticache:*:*:replicationgroup:*",
            "arn:aws:elasticache:*:*:serverlesscache:*"
        ],
        "Condition": {
            "NumericGreaterThanEquals": {
                "elasticache:SnapshotRetentionLimit": "30"
            }
        }
    }
}
]
}

```

17.elasticacheKmsKeyId: especifique o uso de chaves AWS KMS gerenciadas pelo cliente.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ]
        }
    ]
}

```

```

    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateServerlessCache"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:KmsKeyId": "my-key"
        }
    }
}
]
}

```

18. `elasticacheCacheParameterGroupName`: especifique um grupo de parâmetros não padrão com parâmetros específicos de uma organização em seus clusters. Você também pode especificar um padrão de nomenclatura para seus grupos de parâmetros ou exclusão de blocos em um nome de grupo de parâmetros específico. A seguir está um exemplo que restringe o uso de apenas "my-org-param-group".

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [

```

```

        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:CacheParameterGroupName": "my-org-param-group"
        }
    }
}
]
}

```

19. `elasticache:CacheParameterGroupName`: Com o Memcached, especifique um grupo de parâmetros não padrão com parâmetros específicos de uma organização em seus clusters. Você também pode especificar um padrão de nomenclatura para seus grupos de parâmetros ou exclusão de blocos em um nome de grupo de parâmetros específico. A seguir está um exemplo que restringe o uso de apenas "my-org-param-group".

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",

```



```

        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},

{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:CacheParameterGroupName": "my-org-param-group"
        }
    }
}
]
}

```

20elasticache:CreateCacheCluster: Negar a CreateCacheCluster ação se a tag de solicitação Project estiver ausente ou não for igual aDev, ou. QA Prod

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [

```

```

        "elasticache:CreateCacheCluster"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "Null": {
            "aws:RequestTag/Project": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:AddTagsToResource"
    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Project": [
                "Dev",
                "Prod",
                "QA"
            ]
        }
    }
}
]
}

```

21 elasticache:CacheNodeType: Permitindo CreateCacheCluster com cacheNodeType cache.r5.large ou cache.r6g.4xlarge e tag. Project=XYZ

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ]
        }
    ]
}

```

```

    ],
    "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "StringEqualsIfExists": {
            "elasticache:CacheNodeType": [
                "cache.r5.large",
                "cache.r6g.4xlarge"
            ]
        },
        "StringEquals": {
            "aws:RequestTag/Project": "XYZ"
        }
    }
}
]
}

```

22elasticache:CacheNodeType: Permitindo CreateCacheCluster com cacheNodeType cache.r5.large ou cache.r6g.4xlarge e tag. Project=XYZ

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",

```

```

        "arn:aws:elasticache:*:*:subnetgroup:*"
    ],
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateCacheCluster"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:cluster:*"
        ],
        "Condition": {
            "StringEqualsIfExists": {
                "elasticache:CacheNodeType": [
                    "cache.r5.large",
                    "cache.r6g.4xlarge"
                ]
            },
            "StringEquals": {
                "aws:RequestTag/Project": "XYZ"
            }
        }
    }
]
}

```

Note

Ao criar políticas para impor tags e outras chaves de condição juntas, o `IfExists` condicional pode ser necessário na condição de elementos-chave devido aos requisitos extras de política `elasticache:AddTagsToResource` para solicitações de criação com o parâmetro `--tags`.

Uso de funções vinculadas ao serviço para o Amazon ElastiCache

O Amazon ElastiCache usa [funções vinculadas ao serviço](#) do AWS Identity and Access Management IAM. A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a um serviço da AWS, como o Amazon ElastiCache. As funções vinculadas ao serviço do Amazon

ElastiCache são predefinidas pelo Amazon ElastiCache. Elas incluem todas as permissões que o serviço exige para chamar os serviços da AWS em nome dos seus clusters.

Uma função vinculada ao serviço facilita a configuração do Amazon ElastiCache porque você não precisa adicionar as permissões necessárias manualmente. As funções ainda existem dentro de sua contada AWS mas estão vinculadas aos casos de uso do Amazon ElastiCache e têm permissões predefinidas. Somente o Amazon ElastiCache pode assumir essas funções e somente essas funções podem usar a política de permissões predefinidas. Você pode excluir os perfis somente depois de primeiro excluir seus recursos relacionados. Isso protege seus recursos do Amazon ElastiCache, pois você não pode remover por engano as permissões necessárias para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que contenham Sim na coluna Função vinculada a serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Sumário

- [Permissões de função vinculada ao serviço para o Amazon ElastiCache](#)
 - [Permissões para criar uma função vinculada ao serviço](#)
- [Criação de uma função vinculada ao serviço \(IAM\)](#)
 - [Criação de uma função vinculada ao serviço \(console do IAM\)](#)
 - [Criação de uma função vinculada ao serviço \(CLI do IAM\)](#)
 - [Criação de uma função vinculada ao serviço \(API do IAM\)](#)
- [Edição da descrição de uma função vinculada a serviço para o Amazon ElastiCache](#)
 - [Edição da descrição de uma função vinculada ao serviço \(console do IAM\)](#)
 - [Edição da descrição de uma função vinculada ao serviço \(CLI do IAM\)](#)
 - [Edição da descrição de uma função vinculada ao serviço \(API do IAM\)](#)
- [Exclusão de uma função vinculada ao serviço do Amazon ElastiCache](#)
 - [Limpar uma função vinculada ao serviço](#)
 - [Exclusão de uma função vinculada ao serviço \(console do IAM\)](#)
 - [Exclusão de uma função vinculada ao serviço \(CLI do IAM\)](#)
 - [Exclusã de uma função vinculada ao serviço \(API do IAM\)](#)

Permissões de função vinculada ao serviço para o Amazon ElastiCache

Permissões para criar uma função vinculada ao serviço

Para permitir que uma entidade do IAM crie uma função vinculada ao serviço `ServiceRoleForElastiCache` da AWS

Adicione a seguinte declaração de política às permissões dessa entidade 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"}}
}
```

Para permitir que uma entidade do IAM exclua uma função vinculada ao serviço `ServiceRoleForElastiCache` da AWS

Adicione a seguinte declaração de política às permissões dessa entidade 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"}}
}
```

Como alternativa, você pode usar uma política gerenciada pela AWS para fornecer acesso total ao Amazon ElastiCache.

Criação de uma função vinculada ao serviço (IAM)

Você pode criar uma função vinculada ao serviço usando o console do IAM, a CLI ou a API.

Criação de uma função vinculada ao serviço (console do IAM)

Você pode usar o console do IAM para criar uma função vinculada ao serviço.

Para criar uma função vinculada ao serviço (console)

1. Faça login em Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Perfis. Em seguida, escolha Criar nova função.
3. Em Select type of trusted entity (Selecionar tipo de entidade confiável), selecione Serviço da AWS.
4. Em Or select a service to view its use cases (Ou selecione um serviço para visualizar seus casos de uso), escolha ElastiCache.
5. Escolha Next: Permissions (Próximo: permissões).
6. Em Nome da política, observe que `ElastiCacheServiceRolePolicy` é necessário para esta função. Escolha Próximo: tags.
7. Observe que não há suporte para as tags para funções vinculadas ao serviço. Escolha Próximo: análise.
8. (Opcional) Em Descrição da função, edite a descrição para a nova função vinculada ao serviço.
9. Revise a função e escolha Criar função.

Criação de uma função vinculada ao serviço (CLI do IAM)

Você pode usar operações do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço. Essa função pode incluir a política de confiança e as políticas em linha de que o serviço precisa para assumir a função.

Para criar uma função vinculada ao serviço (CLI)

Use a seguinte operação:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

Criação de uma função vinculada ao serviço (API do IAM)

Você pode usar a API do IAM para excluir uma função vinculada ao serviço. Essa função pode conter a política de confiança e as políticas em linha de que o serviço precisa para assumir a função.

Para criar uma função vinculada ao serviço (API)

Use a chamada da API [CreateServiceLinkedRole](#). Na solicitação, especifique o nome do serviço na forma de `elasticache.amazonaws.com`.

Edição da descrição de uma função vinculada a serviço para o Amazon ElastiCache

O Amazon ElastiCache não permite que você edite a função vinculada ao serviço `ServiceRoleForElastiCache` da AWS. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM.

Edição da descrição de uma função vinculada ao serviço (console do IAM)

Também é possível usar o console do IAM para editar a descrição de uma função vinculada ao serviço.

Para editar a descrição de uma função vinculada ao serviço (console)

1. No painel de navegação do console do IAM, escolha Perfis.
2. Escolha o nome da função a ser modificada.
3. No extremo direito da Descrição da função, escolha Editar.
4. Insira uma nova descrição na caixa e escolha Salvar.

Edição da descrição de uma função vinculada ao serviço (CLI do IAM)

Você pode usar operações do IAM na AWS Command Line Interface para editar a descrição de uma função vinculada ao serviço.

Para alterar a descrição de uma função (CLI)

1. (Opcional) Para visualizar a descrição atual de uma função, use a AWS CLI para a operação do IAM [get-role](#).

Example

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

Use o nome da função, não o nome de recurso da Amazon (ARN), para fazer referência às funções com as operações da CLI. Por exemplo, se uma função tiver o seguinte nome de

recurso da Amazon (ARN): `arn:aws:iam::123456789012:role/myrole`, você fará referência à função como **myrole**.

2. Para atualizar a descrição de uma função vinculada ao serviço, use a AWS CLI para a operação do IAM [update-role-description](#).

Para Linux, macOS ou Unix:

```
$ aws iam update-role-description \  
    --role-name AWSServiceRoleForElastiCache \  
    --description "new description"
```

Para Windows:

```
$ aws iam update-role-description ^  
    --role-name AWSServiceRoleForElastiCache ^  
    --description "new description"
```

Edição da descrição de uma função vinculada ao serviço (API do IAM)

Você pode usar a API do IAM para editar uma descrição de função vinculada ao serviço.

Para alterar a descrição de uma função (API)

1. (Opcional) Para visualizar a descrição atual de uma função, use a operação da API do IAM [GetRole](#).

Example

```
https://iam.amazonaws.com/  
?Action=GetRole  
&RoleName=AWSServiceRoleForElastiCache  
&Version=2010-05-08  
&AUTHPARAMS
```

2. Para atualizar uma descrição de função, use a operação da API do IAM [UpdateRoleDescription](#).

Example

```
https://iam.amazonaws.com/  
?Action=UpdateRoleDescription
```

```
&RoleName=AWSServiceRoleForElastiCache
&Version=2010-05-08
&Description="New description"
```

Exclusão de uma função vinculada ao serviço do Amazon ElastiCache

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar sua função vinculada ao serviço antes de excluí-la.

O Amazon ElastiCache não exclui a função vinculada ao serviço para você.

Limpar uma função vinculada ao serviço

Antes de usar o IAM para excluir uma função vinculada ao serviço, primeiro confirme que a função não tem recursos (clusters ou grupos de replicação) associados a ela.

Para verificar se a função vinculada ao serviço tem uma sessão ativa no console do IAM

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Perfis. Então, escolha o nome (não a caixa de seleção) da função ServiceRoleForElastiCache da AWS.
3. Na página Resumo para a função selecionada, escolha a guia Consultor de Acesso.
4. Na guia Consultor de Acesso, revise a atividade recente para a função vinculada ao serviço.

Para excluir recursos do Amazon ElastiCache que exigem ServiceRoleForElastiCache da AWS

- Para excluir um cluster, consulte o seguinte:
 - [Usando o Console de gerenciamento da AWS](#)
 - [Usando o AWS CLI para excluir um ElastiCache cluster](#)
 - [Usando a ElastiCache API](#)
- Para excluir um grupo de replicação, consulte o seguinte:
 - [Exclusão de um grupo de replicação \(Console\)](#)
 - [Excluir um grupo de replicação \(AWS CLI\)](#)
 - [Exclusão de um grupo de replicação \(API do ElastiCache\)](#)

Exclusão de uma função vinculada ao serviço (console do IAM)

É possível usar o console do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (console)

1. Faça login em Console de gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Roles. Selecione a caixa de marcação ao lado do nome da função que você deseja excluir, não o nome ou a linha em si.
3. Em ações de Role (Função) na parte superior da página, escolha a função Delete (Excluir).
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço, que mostram quando cada uma das funções selecionadas acessou pela última vez um serviço da AWS. Isso ajuda você a confirmar se a função está ativo no momento. Se quiser prosseguir, escolha Sim, Excluir para enviar a função vinculada ao serviço para exclusão.
5. Monitore as notificações do console do IAM para progresso da exclusão da função vinculada ao serviço. Como a exclusão da função vinculada ao serviço do IAM é assíncrona, depois de enviar a função para exclusão, a tarefa pode ou não ser bem-sucedida. Se a tarefa obtiver êxito, você poderá escolher Visualizar Detalhes ou Visualizar Recursos a partir das notificações para saber por que a exclusão falhou.

Exclusão de uma função vinculada ao serviço (CLI do IAM)

Você pode usar operações do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço.

Para excluir uma função vinculado ao serviço (CLI)

1. Se você não souber o nome da função vinculada ao serviço que deseja excluir, insira o seguinte comando. Esse comando lista as funções e os nomes de recursos da Amazon (ARNs) em sua conta.

```
$ aws iam get-role --role-name role-name
```

Use o nome da função, não o nome de recurso da Amazon (ARN), para fazer referência às funções com as operações da CLI. Por exemplo, se uma função tiver o ARN `arn:aws:iam::123456789012:role/myrole`, você fará referência à função como **myrole**.

2. Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `deletion-task-id` da resposta para verificar o status da tarefa de exclusão. Insira o seguinte para enviar uma solicitação de exclusão de função vinculada ao serviço.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Insira o seguinte para verificar o estado da tarefa de exclusão.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Exclusã de uma função vinculada ao serviço (API do IAM)

É possível usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (API)

1. Para enviar uma solicitação de exclusão de um roll vinculada ao serviço, chame [DeleteServiceLinkedRole](#). Na solicitação, especifique o nome da função.

Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `DeletionTaskId` da resposta para verificar o status da tarefa de exclusão.

2. Para verificar o status da exclusão, chame [GetServiceLinkedRoleDeletionStatus](#). Na solicitação, especifique o `DeletionTaskId`.

O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Permissões da API do ElastiCache: referência de ações, recursos e condições

Ao configurar [controle de acesso](#) e escrever políticas de permissões para anexar a uma política do IAM (baseada em identidade ou em recursos), use a tabela a seguir como referência. A tabela lista cada operação da API do Amazon ElastiCache e as ações correspondentes para as quais você pode conceder permissões para executar a ação. Você especifica as ações no campo `Action` da política e um valor de recurso no campo `Resource` da política. Salvo indicação em contrário, o recurso é obrigatório. Alguns campos incluem um recurso obrigatório e recursos opcionais. Quando não há ARN de recurso, o recurso na política é um caractere curinga (*).

Você pode usar as chaves de condição em toda a nas suas políticas do ElastiCache para expressar condições. Para ver uma lista de chaves de condição específicas do ElastiCache, juntamente com as ações e os tipos de recursos aos quais elas se aplicam, consulte [Uso de chaves de condição](#). Para obter uma lista completa de chaves em toda a AWS, consulte [Chaves de contexto de condição globais da AWS](#) no Guia do usuário do IAM.

Note

Para especificar uma ação, use o prefixo `elasticache:` seguido do nome da operação da API (por exemplo, `elasticache:DescribeCacheClusters`).

Para ver uma lista das ações do ElastiCache, consulte [Actions Defined by Amazon ElastiCache](#) na Referência de autorização do serviço.

Validação de conformidade para a Amazon ElastiCache

Audidores terceirizados avaliam a segurança e a conformidade dos AWS serviços como parte de vários programas de AWS conformidade, como SOC, PCI, FedRAMP e HIPAA.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a [documentação AWS de segurança](#).

- Quando a Amazon ElastiCache é listada como estando dentro do escopo de um programa de conformidade, esse escopo se aplica a todos os mecanismos de cache suportados pela ElastiCache. Isso inclui ElastiCache para Valkey, ElastiCache Memcached e ElastiCache Redis OSS.

Mais informações

Para obter informações gerais sobre a conformidade com a AWS nuvem, consulte o seguinte:

- [Endpoints FIPS por serviço](#)
- [Atualizações de serviço em ElastiCache](#)
- [AWS Conformidade na nuvem](#)
- [Modelo de responsabilidade compartilhada](#)
- [AWS Programa de conformidade com o PCI DSS](#)

Resiliência na Amazon ElastiCache

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Além da infraestrutura AWS global, a Amazon ElastiCache oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

Tópicos

- [Atenuar falhas](#)

Atenuar falhas

Ao planejar sua ElastiCache implementação na Amazon, você deve planejar de forma que as falhas tenham um impacto mínimo em seus aplicativos e dados. Os tópicos nesta seção discutem as abordagens que você pode tomar para proteger seu aplicativo e dados contra falhas.

Tópicos

- [Mitigar falhas ao executar o Memcached](#)
- [Mitigar falhas ao executar o Valkey ou Redis OSS](#)
- [Recomendações](#)

Mitigar falhas ao executar o Memcached

Ao executar o mecanismo Memcached, você tem as seguintes opções para minimizar o impacto de uma falha. Há dois tipos de falhas que devem ser resolvidas em seus planos de mitigação de falhas: falhas de nó e falhas na zona de disponibilidade.

Mitigar falhas de nós

Os caches sem servidor mitigam automaticamente falhas em nó com uma arquitetura multi-AZ, de maneira que as falhas no nó sejam transparentes para a aplicação. Para mitigar o impacto de uma falha de nó em um cluster baseado em nós, espalhe seus dados em cache por mais nós. Como os clusters baseados em nós não oferecem suporte para replicação, uma falha de nó sempre resultará em uma certa perda de dados do seu cluster.

Ao criar seu cluster do Memcached, você pode criá-lo com 1 a 60 nós, ou mais, por solicitação especial. Particionar seus dados em um número maior de nós significa que você perderá menos dados se um nó falhar. Por exemplo, se você particionar seus dados em 10 nós, um único nó armazenará aproximadamente 10% dos seus dados em cache. Nesse caso, uma falha de nó perde aproximadamente 10% do seu cache, que precisa ser substituído quando um nó de substituição é criado e provisionado. Se os mesmos dados tiverem sido armazenados em cache em 3 nós maiores, a falha de um nó perderia aproximadamente 33% dos seus dados em cache.

Para obter informações sobre como especificar o número de nós em um cluster Memcached, consulte [Criação de um cluster do Memcached \(console\)](#).

Mitigar falhas de zonas de disponibilidade

Os caches sem servidor mitigam automaticamente falhas em zona de disponibilidade com uma arquitetura multi-AZ replicada, de maneira que as falhas em AZ sejam transparentes para a aplicação.

Para mitigar o impacto de uma falha na zona de disponibilidade em um cluster baseado em nós, localize seus nós no maior número possível de zonas de disponibilidade. No caso improvável de uma falha de AZ, você perderá os dados armazenados em cache nessa AZ, não os dados em cache na outra. AZs

Por que tantos nós?

Se a minha região tem apenas três zonas de disponibilidade, por que preciso de mais de três nós, já que, se uma AZ falhar, perco aproximadamente um terço dos meus dados?

Esta é uma excelente pergunta. Lembre-se de que estamos tentando mitigar dois tipos distintos de falhas: de nó e de zona de disponibilidade. Você está certo. Se os seus dados estiverem espalhados por zonas de disponibilidade e uma delas falhar, você perderá apenas os dados armazenados em cache naquela AZ, independentemente do número de nós que tem. No entanto, se um nó falhar, ter mais nós reduzirá a proporção de dados perdidos.

Não há uma "fórmula mágica" para determinar quantos nós você tem no seu cluster. Você deve ponderar o impacto da perda de dados versus a probabilidade de uma falha versus custo e chegar à sua própria conclusão.

Para obter informações sobre como especificar o número de nós em um cluster Memcached, consulte [Criação de um cluster do Memcached \(console\)](#).

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#).

Mitigar falhas ao executar o Valkey ou Redis OSS

Ao executar um mecanismo Valkey ou Redis OSS, você tem as opções a seguir para minimizar o impacto de um nó ou falha da zona de disponibilidade.

Mitigar falhas de nós

Os caches sem servidor mitigam automaticamente falhas em nó com uma arquitetura multi-AZ, de maneira que as falhas no nó sejam transparentes para a aplicação. Os clusters baseados em nós devem ser devidamente configurados para mitigar a falha de um nó individual.

Para mitigar o impacto de falhas em nós do Valkey ou Redis OSS em clusters baseados em nós, você tem as seguintes opções:

Tópicos

- [Mitigar falhas: grupos de replicação do Valkey ou Redis OSS](#)

Mitigar falhas: grupos de replicação do Valkey ou Redis OSS

Um grupo de replicação do Valkey ou Redis OSS é composto por um único nó primário no/do qual sua aplicação pode ler e gravar e de 1 a 5 nós de réplica somente leitura. Sempre que os dados são gravados no nó primário, eles também são atualizados de maneira assíncrona nos nós de réplica de leitura.

Quando uma réplica de leitura falha

1. ElastiCache detecta a falha na réplica de leitura.
2. ElastiCache desativa o nó com falha.
3. ElastiCache lança e provisiona um nó de substituição na mesma AZ.
4. O novo nó é sincronizado com o nó primário.

Durante esse período, seu aplicativo pode continuar lendo e gravando usando os outros nós.

Multi-AZ do Valkey ou do Redis OSS

É possível habilitar o Multi-AZ em seus grupos de replicação do Valkey ou Redis OSS. Independentemente de você habilitar o Multi-AZ, uma falha primária será detectada e substituída automaticamente. Como isso acontece depende de o recurso Multi-AZ estar ou não habilitado.

Quando Multi-AZ está habilitado

1. ElastiCache detecta a falha do nó primário.
2. ElastiCache promove o nó de réplica de leitura com o menor atraso de replicação em relação ao nó primário.
3. As outras réplicas se sincronizam com o novo nó primário.
4. ElastiCache ativa uma réplica de leitura no AZ do primário que falhou.
5. O novo nó é sincronizado com o primário recém-promovido.

O failover em um nó de réplica geralmente é mais rápido do que criar e provisionar um novo nó primário. Isso significa que seu aplicativo pode retomar a gravação no nó primário mais cedo do que se o Multi-AZ não estivesse habilitado.

Para obter mais informações, consulte [Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#).

Quando o Multi-AZ está desabilitado

1. ElastiCache detecta falha primária.
2. ElastiCache coloca o primário offline.
3. ElastiCache cria e provisiona um novo nó primário para substituir o primário que falhou.
4. ElastiCache sincroniza o novo primário com uma das réplicas existentes.
5. Quando a sincronização é concluída, o novo nó funciona como nó primário do cluster.

Durante as etapas de 1 a 4 desse processo, seu aplicativo não pode gravar no nó primário. No entanto, seu aplicativo pode continuar a ler dos seus nós de réplica.

Para maior proteção, recomendamos que você inicie os nós do seu grupo de replicação em diferentes zonas de disponibilidade (AZs). Se você fizer isso, uma falha de AZ afetará apenas os nós nessa AZ, e não os outros.

Para obter mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

Mitigar falhas de zonas de disponibilidade

Os caches sem servidor mitigam automaticamente falhas em zona de disponibilidade com uma arquitetura multi-AZ replicada, de maneira que as falhas em AZ sejam transparentes para a aplicação.

Para mitigar o impacto de uma falha na zona de disponibilidade em um cluster baseado em nós, localize seus nós para cada fragmento no maior número possível de zonas de disponibilidade.

Não importa quantos nós você tenha em um fragmento, se todos estiverem localizados na mesma zona de disponibilidade, uma falha catastrófica dessa AZ resultará na perda de todos os dados do fragmento. No entanto, se você localizar seus nós em vários AZs, uma falha em qualquer AZ resultará na perda somente dos nós nessa AZ.

Sempre que você perde um nó, pode passar por uma degradação do desempenho, pois as operações de leitura são compartilhadas por menos nós. Essa degradação do desempenho continuará até que os nós sejam substituídos.

Para obter informações sobre como especificar as zonas de disponibilidade para nós do Valkey ou do Redis OSS, consulte [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#).

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Seleção de regiões e zonas de disponibilidade para o ElastiCache](#).

Recomendações

É recomendável criar caches com tecnologia sem servidor em clusters baseados em nós, pois você obtém automaticamente uma melhor tolerância a falhas sem configuração adicional. No entanto, durante a criação de um cluster baseado em nós, existem dois tipos de falhas para as quais você precisa se planejar: falhas de nós individuais e falhas amplas de zona de disponibilidade. O melhor plano de mitigação de falhas abordará ambos os tipos de falhas.

Minimizar o impacto das falhas em nó

Para minimizar o impacto de uma falha de nó ao usar o Valkey ou o Redis OSS, recomendamos que sua implementação use vários nós em cada fragmento e distribua os nós em várias zonas de disponibilidade. Isso é feito automaticamente para caches sem servidor.

Para clusters baseados em nós no Valkey ou no Redis OSS, recomendamos que você habilite o Multi-AZ em seu grupo de replicação para que ele passe automaticamente para uma réplica ElastiCache se o nó primário falhar.

Quando estiver executando o Memcached e particionando seus dados entre nós, quanto mais nós usar, menor será a perda de dados se qualquer um dos nós falhar.

Minimizar o impacto das falhas na zona de disponibilidade

Para minimizar o impacto de uma falha na zona de disponibilidade, nós recomendamos a execução dos nós em quantas zonas de disponibilidade diferentes forem disponíveis. Distribuir seus nós uniformemente AZs minimizará o impacto no caso improvável de uma falha no AZ. Isso é feito automaticamente para caches sem servidor.

Outras precauções

Se você estiver executando o Valkey ou o Redis OSS, além da precaução acima, recomendamos que você programe backups regulares do seu cluster. Backups (snapshots) criam um arquivo .rdb

que você pode usar para restaurar seu cache em caso de falha ou corrupção. Para obter mais informações, consulte [Snapshots e restauração](#).

Segurança da infraestrutura no ElastiCache da AWS

Como um serviço gerenciado, o ElastiCache da AWS é protegido pelos procedimentos de segurança de rede global da AWS descritos na seção Segurança e conformidade do [Centro de Arquitetura da AWS](#).

Você usa chamadas de API publicadas pela AWS para acessar o ElastiCache por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem ter compatibilidade com conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece compatibilidade com esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Atualizações de serviço em ElastiCache

ElastiCache monitora automaticamente sua frota de caches, clusters e nós para aplicar atualizações de serviço à medida que elas se tornam disponíveis. As atualizações de serviço para caches sem servidor são aplicadas de maneira transparente e automática. Para clusters baseados em nós, você configura uma janela de manutenção predefinida para ElastiCache poder aplicar essas atualizações. No entanto, em alguns casos, você pode achar que essa abordagem é muito rígida e que provavelmente restringirá os fluxos de negócios.

Com as atualizações de serviço, você pode controlar quando e quais atualizações são aplicadas aos clusters baseados em nós. Você também pode monitorar o progresso dessas atualizações no ElastiCache cluster selecionado em tempo real.

Tópicos

- [Gerenciamento de atualizações de serviço para clusters baseados em nós](#)

Gerenciamento de atualizações de serviço para clusters baseados em nós

ElastiCache atualizações de serviço para clusters baseados em nós são lançadas regularmente. Se você tiver um ou mais clusters qualificados baseados em nós para essas atualizações de serviço, receberá notificações por e-mail, SNS, Personal Health Dashboard (PHD) e eventos da CloudWatch Amazon quando as atualizações forem lançadas. As atualizações também são exibidas na página Atualizações de serviços no ElastiCache console. Ao usar esse painel, você pode visualizar todas as atualizações de serviço e o status de sua ElastiCache frota. As atualizações de serviço para caches sem servidor são aplicadas de maneira transparente e não podem ser gerenciadas por meio de Atualizações de serviço.

Você controla quando aplicar uma atualização antes do início da atualização automática. É altamente recomendável que você aplique qualquer atualização do tipo security-update o mais rápido possível para garantir que seus ElastiCache clusters estejam sempre up-to-date com os patches de segurança atuais.

As seguintes seções analisam essas opções em detalhes.

Como aplicar as atualizações de serviço

Será possível começar a aplicar as atualizações de serviços à sua frota desde o momento em que as atualizações tiverem um status disponível. As atualizações de serviço são cumulativas. Em outras palavras, qualquer atualização que você ainda não tiver aplicado serão incluídas na sua atualização mais recente.

Se uma atualização de serviço tiver a atualização automática ativada, você poderá optar por anotar qualquer ação quando ela estiver disponível. ElastiCache agendará a aplicação da atualização durante uma das próximas janelas de manutenção de seus clusters após a data de início da atualização automática. Você receberá notificações relacionadas a cada etapa da atualização.

Note

É possível aplicar somente as atualizações de serviço que tenham um status disponível ou programado.

Para obter mais informações sobre como revisar e aplicar quaisquer atualizações específicas do serviço aos ElastiCache clusters aplicáveis, consulte [Aplicação de atualizações de serviço usando o console](#)

Quando uma nova atualização de serviço está disponível para um ou mais de seus ElastiCache clusters, você pode usar o ElastiCache console, a API ou AWS CLI aplicar a atualização. As seções a seguir explicam as opções que você pode usar para aplicar as atualizações.

Aplicação de atualizações de serviço usando o console

Para visualizar a lista de atualizações de serviço disponíveis, além de outras informações, acesse a página Atualizações de serviço no console.

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione Service Updates (Atualizações de serviço).
3. Em Service Updates (Atualizações de serviço), é possível visualizar o seguinte:
 - Nome da atualização de serviço: o nome exclusivo da atualização de serviço
 - Tipo de atualização: o tipo da atualização de serviço, que é security-update (atualização-de-segurança) ou engine-update (atualização-de-mecanismo)
 - Update Severity (Gravidade da atualização): a prioridade de aplicação da atualização:
 - crítica: recomendamos que você aplique essa atualização imediatamente (dentro de 14 dias ou menos).
 - importante: recomendamos que você aplique essa atualização assim que seu fluxo de negócios permitir (dentro de 30 dias ou menos).
 - média: recomendamos que você aplique essa atualização assim que possível (dentro de 60 dias ou menos).
 - baixa: recomendamos que você aplique essa atualização assim que possível (dentro de 90 dias ou menos).
 - Versão do mecanismo: se o tipo de atualização for atualização do mecanismo, a versão do mecanismo que está sendo atualizada.
 - Data de lançamento: quando a atualização está liberada e disponível para aplicação nos clusters.
 - Recommended Apply By Date (Aplicação recomendada até a data): data de orientação do ElastiCache até a qual as atualizações devem ser aplicadas.
 - Status: o status da atualização, que pode ser um dos seguintes:
 - disponível: a atualização está disponível para clusters de requisito.
 - concluída: a atualização foi aplicada.

- cancelada: a atualização foi cancelada e não é mais necessária.
 - expired (expirada): a atualização não está mais disponível para ser aplicada.
4. Escolha uma atualização individual (não o botão à esquerda) para visualizar os detalhes da atualização do serviço.

Em Cluster update status (Status da atualização do cluster), é possível visualizar uma lista de clusters nos quais a atualização do serviço não foi aplicada ou acabou de ser aplicada recentemente. Para cada cluster, é possível visualizar o seguinte:

- Nome do cluster: o nome do cluster
- Nós atualizados: a proporção de nós individuais dentro de um cluster específico que foram atualizados ou permanecem disponíveis para a atualização de serviço específica.
- Tipo de atualização: o tipo da atualização de serviço, que é security-update (atualização-de-segurança) ou engine-update (atualização-de-mecanismo)
- Status: o status da atualização de serviço no cluster, que é um dos seguintes:
 - disponível: a atualização está disponível para clusters de requisito.
 - em andamento: a atualização está sendo aplicada a esse cluster.
 - programada: a data de atualização foi programada.
 - concluída: a atualização foi aplicada com êxito. O cluster com status completo será exibido por 7 dias após sua conclusão.

Se você escolheu qualquer um ou todos os clusters com o status disponível ou programado e, em seguida, escolheu Aplicar agora, a atualização começará a ser aplicada nesses clusters.

Aplicação das atualizações de serviços usando a AWS CLI

Depois de receber a notificação de que há atualizações de serviços disponíveis, você poderá inspecioná-las e aplicá-las usando a AWS CLI:

- Para recuperar uma descrição das atualizações de serviços disponíveis, execute o seguinte comando:

```
aws elasticache describe-service-updates --service-update-status available
```

Para obter mais informações, consulte [describe-service-updates](#).

- Para aplicar uma atualização de serviço em uma lista de clusters, execute o seguinte comando:

```
aws elasticache batch-apply-update-action --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

Para obter mais informações, consulte [batch-apply-update-action](#).

Verificando se você tem a atualização de serviço mais recente aplicada usando o console AWS

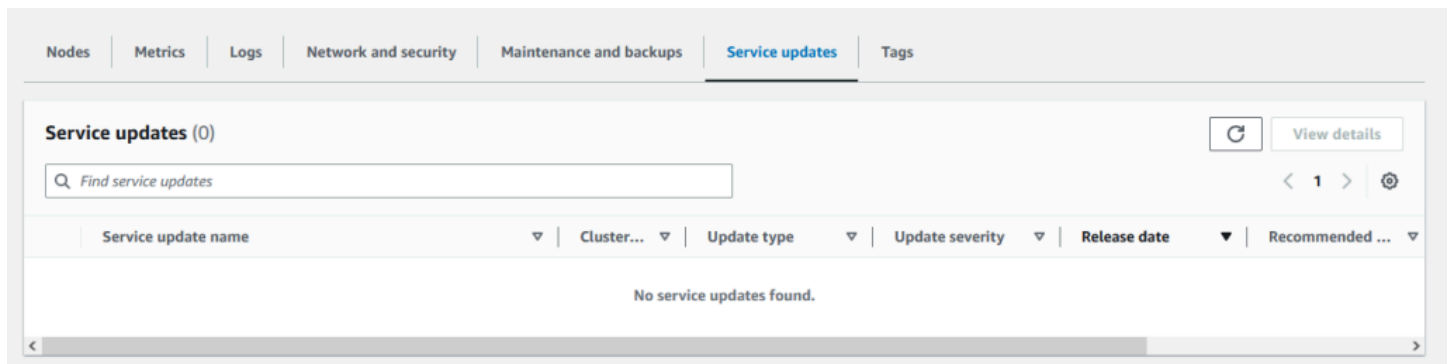
Você pode verificar se seus ElastiCache clusters do Redis OSS estão executando a atualização de serviço mais recente seguindo estas etapas:

- Escolha um cluster aplicável na página Redis OSS Clusters
- Escolha Atualizações de serviço no painel de navegação para ver as atualizações de serviço aplicáveis para esse cluster, se houver.

Se o console exibir uma lista de atualizações de serviço, você poderá selecionar a atualização de serviço e escolher Aplicar agora.

Service updates (3)									
Find service updates									
Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-b...	Status	Cluster ...		
☐ elasticache-redis-6-2-6-update-20230109	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, 03:59:59 (UT...	Available	March 1, 2...		

Se o console exibir “Nenhuma atualização de serviço encontrada”, significa que o cluster OSS ElastiCache for Redis já tem a atualização de serviço mais recente aplicada.



Interromper as atualizações do serviço

É possível interromper as atualizações em clusters, se necessário. Por exemplo, será possível interromper as atualizações se houver uma sobrecarga inesperada nos clusters que estão recebendo atualizações. Ou pode interrompê-las se elas estiverem demorando muito e interrompendo o fluxo de negócios em um horário de pico.

A operação [Stopping \(Interrompendo\)](#) interrompe automaticamente todas as atualizações desses clusters e de quaisquer nós que ainda serão atualizados. Ela continua até a conclusão dos nós que tiverem um status in progress (em andamento). Porém, ela cessa as atualizações em outros nós do mesmo cluster que tenham um status update available (atualização disponível) e as reverte para um status Stopping (Interrompendo).

Quando o fluxo de trabalho Stopping (Interrompendo) for concluído, os nós que tiverem um status Stopping (Interrompendo) serão alterados para um status Stopped (Interrompido). Dependendo do fluxo de trabalho da atualização, alguns clusters não terão nenhum nó atualizado. Outros clusters poderão incluir alguns nós que foram atualizados e outros que ainda terão um status update available (atualização disponível).

Você poderá voltar mais tarde para concluir o processo de atualização conforme seus fluxos de negócios permitirem. Nesse caso, escolha os clusters aplicáveis em que deseja concluir as atualizações e selecione Apply Now (Aplicar agora). Para obter mais informações, consulte [Como aplicar as atualizações de serviço](#).

Usar o console do

Você pode interromper uma atualização de serviço usando o ElastiCache console. A seguir é demonstrado como fazer isso:

- Depois que uma atualização de serviço progride em um cluster selecionado, o ElastiCache console exibe a guia Exibir/Interromper Atualização na parte superior do painel. ElastiCache

- Para interromper a atualização, escolha Stop Update (Interromper atualização).
- Quando você interromper a atualização, escolha o cluster e examine o status. Ela é revertida para o status Interrompendo e, por fim, para o status Interrompida.

Usando o AWS CLI

Você pode interromper uma atualização de serviço usando a AWS CLI. O exemplo de código a seguir mostra como fazer isso.

Para um grupo de replicação, faça o seguinte:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

Para um cluster, faça o seguinte:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

Para obter mais informações, consulte [BatchStopUpdateAction](#).

Vulnerabilidades e exposições comuns (CVE): vulnerabilidades de segurança abordadas em ElastiCache

A lista de Vulnerabilidades e exposições comuns (CVE) é uma lista de vulnerabilidades de segurança cibernética publicamente conhecidas. Cada entrada é um link que contém um número de identificação, uma descrição e pelo menos uma referência pública. Você pode encontrar nesta página uma lista de vulnerabilidades de segurança que foram abordadas em ElastiCache.

Recomendamos que você sempre atualize para as versões mais recentes do ElastiCache Valkey, Redis OSS ou ElastiCache Memcached para se proteger contra vulnerabilidades conhecidas. Ao operar um cache ElastiCache sem servidor, as correções de CVE são aplicadas automaticamente ao seu cache. Ao operar clusters baseados em nós com Valkey ou Redis OSS, ElastiCache expõe o componente PATCH. Por exemplo, ao usar ElastiCache para o Redis OSS versão 6.2.6, a versão principal é 6, a versão secundária é 2 e a versão do patch é 6. As versões do PATCH são para correções de bugs compatíveis com versões anteriores, correções de segurança e alterações não funcionais.

Você pode usar a tabela a seguir para verificar se uma versão específica do ElastiCache Valkey e do Redis OSS tem uma correção para uma vulnerabilidade de segurança específica. Se seu cluster OSS ElastiCache Valkey ou Redis estiver executando uma versão sem a correção de segurança, consulte a tabela abaixo e tome medidas. Você pode atualizar para uma versão mais recente do ElastiCache Valkey ou do Redis OSS contendo a correção ou, se estiver em uma versão que contenha a correção, verifique se a atualização de serviço mais recente foi aplicada consultando a. [Gerenciamento de atualizações de serviço para clusters baseados em nós](#) Para obter mais informações sobre as versões suportadas ElastiCache do mecanismo e como fazer o upgrade, consulte [Versões e atualizações do motor em ElastiCache](#).

 Note

- Se um CVE for endereçado em uma ElastiCache versão, isso significa que ele também é abordado nas versões mais recentes. Então, por exemplo, se uma vulnerabilidade for resolvida na ElastiCache versão 6.0.5 do Redis OSS, isso continuará nas versões 6.2.6, 7.0.7 e 7.1.
- Um asterisco (*) na tabela a seguir indica que você deve ter a atualização de serviço mais recente aplicada ao cluster do OSS ElastiCache for Redis executando a versão do OSS ElastiCache for Redis especificada para resolver a vulnerabilidade de segurança. Para obter mais informações sobre como verificar se você tem a atualização de serviço mais recente aplicada ElastiCache para a versão do Redis OSS em que seu cluster está sendo executado, consulte. [Gerenciamento de atualizações de serviço para clusters baseados em nós](#)

ElastiCache versão	CVEs Endereçado
Valkey 8.1 e todas as versões anteriores do Valkey	CVE-2025-49844 *, CVE-2025-46817 *, CVE-2025-46818 *, CVE-2025-46819 *
Redis OSS 7.1 e todas as versões anteriores do Redis OSS	
Valkey 7.2 e 7.3	CVE-2025-21607 *, CVE-2025-21605 *, CVE-2024-31449 *, CVE-2024-31227 *, CVE-2024-31228 *

ElastiCache versão	CVEs Endereçado
Vale 7.2.7	CVE-2024-51741
Redis OSS 7.1 e 6.2	CVE-2025-21605* , CVE-2024-31449* , CVE-2024-31227* , CVE-2024-31228* , CVE-2023-41056
Redis OSS 7.0.7	CVE-2023-41056*
Redis OSS 6.2.7	CVE-2024-46981
Redis OSS 6.2.6	CVE-2022-24834* , CVE-2022-35977* , CVE-2022-36021* , CVE-2023-22458 , CVE-2023-25155 , CVE-2023-28856 CVE-2023-45145 : observe que esse CVE foi abordado no Redis OSS 6.2 e 7.0, mas não no Redis OSS 7.1.
Redis OSS 6.0.5	CVE-2022-24735* , CVE-2022-24736*

Registros e monitoramento no Amazon ElastiCache

Para gerenciar o cache, é importante que você saiba o desempenho dos caches. O ElastiCache gera métricas publicadas no Amazon CloudWatch Logs para monitorar o desempenho do cache. Além disso, o ElastiCache gera eventos quando mudanças significativas acontecem nos recursos de cache (por exemplo, um novo cache é criado ou um cache é excluído).

Tópicos

- [Métricas e eventos para caches com tecnologia sem servidor para Valkey e Redis OSS](#)
 - [Métricas e eventos de cluster baseados em nós para Valkey e Redis OSS](#)
 - [Métricas e eventos para caches e clusters do Memcached](#)
 - [Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail](#)
 - [Monitoramento de eventos do ElastiCache pelo Amazon SNS](#)
 - [Entrega de logs](#)
 - [Monitorando o uso com CloudWatch métricas](#)
-
- [Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail](#)

Métricas e eventos para caches com tecnologia sem servidor para Valkey e Redis OSS

ElastiCache oferece uma ampla variedade de métricas e eventos para monitoramento ao trabalhar com caches sem servidor. Isso inclui CloudWatch métricas, métricas de nível de comando e registros de eventos que podem ser ingeridos via Amazon EventBridge.

Tópicos

- [Métricas de cache sem servidor](#)
- [Eventos de cache sem servidor](#)

Métricas de cache sem servidor

O AWS/ElastiCache namespace inclui as seguintes CloudWatch métricas para seus caches sem servidor Valkey ou Redis OSS.

Códigos métricos para Valkey ou Redis OSS

Métrica	Description	Unidade
BytesUsedForCache	O número total de bytes usados pelos dados armazenados no cache.	Bytes
ElastiCacheProcessingUnits	O número total de ElastiCacheProcessingUnits (ECPUs) consumido pelas solicitações executadas em seu cache	Contagem
SuccessfulReadRequestLatency	Latência de solicitações de leitura bem-sucedidas.	Microssegundos
SuccessfulWriteRequestLatency	Latência de solicitações de gravação bem-sucedidas	Microssegundos
TotalCmdsCount	Contagem total de todos os comandos executados no cache	Contagem
CacheHitRate	Indica a taxa de acerto do cache. Isso é calculado usando as estatísticas <code>cache_hits</code> e <code>cache_misses</code> da seguinte maneira: $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$.	Percentual
CacheHits	O número de buscas de chaves somente leitura bem-sucedidas no cache.	Contagem
CacheMisses	O número de pesquisas malsucedidas de chaves	Contagem

Métrica	Description	Unidade
	somente para leitura no cache.	
CurrConnections	O número de conexões cliente com o cache.	Contagem
ThrottledCmds	O número de solicitações que foram limitadas ElastiCache porque a carga de trabalho estava aumentando mais rápido do que pode ser escalada. ElastiCache	Contagem
NewConnections	O número total de conexões que foram aceitas pelo servidor durante esse período.	Contagem
CurrItems	O número de itens no cache.	Contagem
CurrVolatileItems	O número de itens no cache com TTL.	Contagem
NetworkBytesIn	Total de bytes transferidos para dentro do cache	Bytes
NetworkBytesOut	Total de bytes transferidos para fora do cache	Bytes
Evictions	A contagem de chaves removidas pelo cache	Contagem

Métrica	Description	Unidade
IamAuthenticationExpirations	O número total de conexões do Valkey ou Redis OSS expiradas autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
IamAuthenticationThrottling	O número total de solicitações AUTH ou HELLO do Valkey ou Redis OSS com controle de utilização autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
KeyAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar chaves que eles não têm permissão para acessar. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
AuthenticationFailures	O número total de tentativas falhadas de autenticação para o Valkey ou Redis OSS usando o comando AUTH. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem

Métrica	Description	Unidade
CommandAuthorizationFailures	O número total de tentativas falhadas por usuários para executar comandos que eles não têm permissão para chamar. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem

Métricas no nível de comando

ElastiCache também emite as seguintes métricas de nível de comando. Para cada tipo de comando, ElastiCache emite a contagem total de comandos e o número de comandos ECPUs consumidos por esse tipo de comando.

Métrica	Description	Unidade
EvalBasedCmds	O número de comandos get recebidos pelo cache.	Contagem
EvalBasedCmdsECPUs	ECPUs consumido por comandos baseados em avaliação.	Contagem
GeoSpatialBasedCmds	O número total de comandos para comandos baseados em dados geoespaciais. Isso é derivado da estatística commandstats do Valkey ou Redis OSS. Ele é derivado somando todos o tipos de comandos geo: geoadd, geodist, geohash, geopos, georadius e georadius bymember.	Contagem

Métrica	Description	Unidade
GeoSpatialBasedCmdsECPUs	ECPUs consumido por comandos baseados em localização geoespacial.	Contagem
GetTypeCmds	O número total de comandos tipo somente leitura. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos do tipo somente leitura (get, hget, scard, lrange etc.)	Contagem
GetTypeCmdsECPUs	ECPUs consumido por comandos de leitura.	Contagem
HashBasedCmds	O número total de comandos baseados em hash. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que atuam em um ou mais hashes (hget, hkeys, hvals, hdel etc.).	Contagem
HashBasedCmdsECPUs	ECPUs consumido por comandos baseados em hash.	Contagem
HyperLogLogBasedCmds	O número total de comandos HyperLogLog baseados. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos do tipo pf (pfadd, pfcount, pfmerge etc.).	Contagem

Métrica	Description	Unidade
HyperLogLogBasedCmdsECPUs	ECPUs consumido por comandos HyperLogLog baseados.	Contagem
JsonBasedCmds	O número total de comandos JSON, incluindo comandos de leitura e gravação. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos JSON que atuam em chaves JSON.	Contagem
JsonBasedCmdsECPUs	ECPUs consumido por todos os comandos JSON, incluindo comandos de leitura e gravação.	Contagem
JsonBasedGetCmds	O número total de comandos JSON somente leitura. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos JSON de leitura que atuam em chaves JSON.	Contagem
JsonBasedGetCmdsECPUs	ECPUs consumido por comandos somente de leitura JSON.	Contagem

Métrica	Description	Unidade
JsonBasedSetCmds	O número total de comandos de gravação JSON. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos JSON de gravação que atuam em chaves JSON.	Contagem
JsonBasedSetCmdsECPUs	ECPUs consumido pelos comandos de gravação JSON.	Contagem
KeyBasedCmds	O número total de comandos baseados em chave. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que atuam em uma ou mais chaves em várias estruturas de dados (del, expire, rename etc.).	Contagem
KeyBasedCmdsECPUs	ECPUs consumido por comandos baseados em teclas.	Contagem
ListBasedCmds	O número total de comandos baseados em lista. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que atuam em uma ou mais listas (lindex, lrange, lpush, ltrim etc.).	Contagem

Métrica	Description	Unidade
ListBasedCmdsECPUs	ECPUs consumido por comandos baseados em listas.	Contagem
NonKeyTypeCmds	O número total de comandos não baseados em chave. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que não atuam em uma chave; por exemplo, acl, dbsize ou info.	Contagem
NonKeyTypeCmdsECPUs	ECPUs consumido por non-key-based comandos.	Contagem
PubSubBasedCmds	O número total de comandos para pub/sub funcionalidade. Isso é derivado das estatísticas de comandos Valkey ou Redis OSS, somando todos os comandos usados para fins de pub/sub funcionalidade: psubscribe, publish, pubsub, punsubscribe, ssubscribe e, sunsubscribe, spublish, subscribe e unsubscribe.	Contagem
PubSubBasedCmdsECPUs	ECPUs consumido por comandos baseados em pub/sub.	Contagem

Métrica	Description	Unidade
SetBasedCmds	O número total de comandos que são baseados em conjuntos. Isso é derivado da estatística commandstats do Valkey ou Redis OSS, somando todos os comandos que atuam em um ou mais conjuntos (scard, sdiff, sadd, sunion etc.).	Contagem
SetBasedCmdsECPUs	ECPUs consumido por comandos baseados em conjuntos.	Contagem
SetTypeCmds	O número total de tipos de comando de gravação. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os tipos de comando mutative que operam em dados (set, hset, sadd, lpop etc.)	Contagem
SetTypeCmdsECPUs	ECPUs consumido por comandos de gravação.	Contagem

Métrica	Description	Unidade
SortedSetBasedCmds	O número total de comandos que são classificados com base em conjuntos. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que atuam em um ou mais conjuntos classificados (zcount, zrange, zrank, zadd etc.).	Contagem
SortedSetBasedCmdsECPUs	ECPUs consumido por comandos baseados em classificação.	Contagem
StringBasedCmds	O número total de comandos baseados em string. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que atuam em uma ou mais strings (strlen, setex, setrange etc.).	Contagem
StringBasedCmdsECPUs	ECPUs consumido por comandos baseados em strings.	Contagem

Métrica	Description	Unidade
StreamBasedCmds	O número total de comandos que são baseados em fluxo. Isso é derivado da estatística commandstats do Valkey ou Redis OSS somando todos os comandos que atuam em um ou mais tipos de dados de fluxos (xrange, xlen, xadd, xdel etc.).	Contagem
StreamBasedCmdsECPUs	ECPUs consumido por comandos baseados em fluxo.	Contagem

Eventos de cache sem servidor

ElastiCache registra eventos relacionados ao seu cache sem servidor. Essas informações incluem a data e a hora do evento, o nome da origem e o tipo de origem do evento, bem como uma descrição do evento. Você pode facilmente recuperar eventos do registro usando o ElastiCache console, o comando AWS CLI `describe-events` ou a ação da API `ElastiCache DescribeEvents`.

Você pode escolher monitorar, ingerir, transformar e agir em ElastiCache eventos usando a Amazon EventBridge. Saiba mais no [guia de primeiros EventBridge passos](#) da Amazon.

Visualizando ElastiCache eventos (Console)

Para visualizar eventos usando o ElastiCache console:

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha Eventos.
3. Na tela Eventos, cada linha da lista representa um evento e exibe a origem do evento, o tipo de evento, o horário GMT do evento e uma descrição do evento. Usando a opção Filtro, você pode especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualizando ElastiCache eventos (AWS CLI)

Para gerar uma lista de ElastiCache eventos usando o AWS CLI, use o comando `describe-events`. Você pode usar parâmetros opcionais para controlar os tipos de eventos listados, o período de tempo dos eventos listados, o número máximo de eventos a serem listados e muito mais.

O código a seguir lista até 40 eventos de cache sem servidor.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

O código a seguir lista todos os eventos de cache sem servidor nas últimas 24 horas (1.440 minutos).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Eventos sem servidor

Esta seção documenta os diferentes tipos de eventos que você pode receber para os caches sem servidor.

Eventos de criação de cache sem servidor

Detail-Type	Description	Unidade	Fonte	Mensagem
Cache criado	Fio de cache	criação	serverless-cache	O cache <cache-name> foi criado e está pronto para ser usado.
Cache criado	Fio de cache Fio de snapshot	criação	serverless-cache	O cache <cache-name> é criado e os dados foram restaurados a partir do snapshot. O cache está pronto para ser usado.

Detail-Type	Description	Unidade	Fonte	Mensagem
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Endereços IP livres insuficientes para criar um endpoint da VPC.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Sub-redes inválidas fornecidas na solicitação.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Limite de cota atingido para criação de um endpoint da VPC.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Você não tem permissões para criar um endpoint da VPC.

Detail-Type	Description	Unidade	Fonte	Mensagem
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Um usuário com uma versão incompatível do Valkey ou do Redis OSS está presente no grupo de usuários < >. user-group-name
Falha na criação de cache	Fio de cache Fio de snapshot do cache	falha	serverless-cache	Falha na criação do cache <cache-name>. O grupo de usuários fornecido < user-group-name > não existe.

Detail-Type	Description	Unidade	Fonte	Mensagem
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Houve falha na restauração de dados a partir do snapshot porque <reason>. Motivos de falha: • falhou na recuperação do arquivo do S3. • O md5 esperado não corresponde ao md5 real. • O arquivo RDB fornecido tem uma versão não compatível.

Eventos de atualização de cache sem servidor (Valkey ou Redis OSS)

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	SecurityGroups atualizado para o cache<cache-name>.

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	Tags atualizadas para o cache <cache-name>.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	Falha na atualização do cache <cache-name>. Um usuário com uma versão incompatível do Valkey ou do Redis OSS está presente no grupo de usuários < >. user-group-name
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do cache. SecurityGroups falha na atualização.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do cache. SecurityGroups a atualização falhou devido a permissões insuficientes.

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do cache. SecurityGroups a atualização falhou porque SecurityGroups eles são inválidos.

Eventos de exclusão de cache sem servidor (Valkey ou Redis OSS)

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Cache excluído	Fio de cache	exclusão	serverless-cache	O cache <cache-name> foi excluído.

Eventos de limite de uso de cache sem servidor (Valkey ou Redis OSS)

Detail-Type	Description	Unidade	Fonte	Mensagem
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	Limites atualizados para o cache <cache-name>.
Limite de cache se aproximando	Fio de cache	notificação	serverless-cache	O slot <X> está usando mais <Y>% do limite por slot de 32 GB. Por exemplo, o slot 10 está usando mais 90% do

Detail-Type	Description	Unidade	Fonte	Mensagem
				limite por slot de 32 GB.
Falha na atualização do cache	Fio de cache	falha	serverless-cache	Falha na atualização de um limite para o cache <cache-name> porque o cache foi excluído.
Falha na atualização do cache	Fio de cache	falha	serverless-cache	Falha na atualização de um limite no cache <cache-name> por causa da configuração inválida.
Falha na atualização do cache	Fio de cache	falha	serverless-cache	Houve falha em uma atualização de limite no cache <cache-name> porque os dados atuais em cache excedem os novos limites. Limpe alguns dados antes de aplicar os limites.

Eventos de snapshot de cache sem servidor (Valkey ou Redis OSS)

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Snapshot criado	Fio de cache Fio de snapshot	criação	serverless-cache-snapshot	Snapshot <snapshot-name> criado para cache <cache-name>.
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	Falha na criação do snapshot para o cache <cache-name>. Falha na criação do snapshot <snapshot-name> com a chave gerenciada pelo cliente <key-id> <reason>. Mensagens de motivo da falha: • porque a chave gerenciada pelo cliente está desabilitada • porque não foi possível encontrar a chave gerenciada pelo cliente

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
				porque a solicitação atingiu o tempo limite
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	Falha na criação do snapshot para o cache <cache-name>. Houve falha na criação do snapshot <snapshot-name> por causa de <reason>. Motivo padrão: por causa de um erro interno
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o instantâneo para o bucket %s porque ElastiCache não tem permissões para o bucket.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque já existe um objeto com o mesmo nome no bucket.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o instantâneo para o bucket '%s' porque o ID da conta do proprietário do bucket foi alterado.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque o bucket do S3 não está acessível.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque o bucket não está acessível.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque o bucket não existe.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' com a chave gerenciada pelo cliente do snapshot de origem %s <reason>.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s'.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na cópia do snapshot	Snapshot arn-1 Snapshot arn-2	falha	serverless-cache-snapshot	Falha na cópia do snapshot <snapshot-name>. Não foi possível copiar o snapshot '%s' para o snapshot '%s' com a chave gerenciada pelo cliente do snapshot de origem <key-id> <reason-name>.
Falha na cópia do snapshot	Snapshot arn-1 Snapshot arn-2	falha	serverless-cache-snapshot	Falha na cópia do snapshot <snapshot-name>. Não foi possível copiar o snapshot '%s' para o snapshot '%s' com a chave gerenciada pelo cliente do instantâneo de destino '%s' '%s'.

Métricas e eventos de cluster baseados em nós para Valkey e Redis OSS

ElastiCache oferece uma ampla variedade de métricas e eventos para monitorar clusters OSS Valkey e Redis baseados em nós. Isso inclui métricas em nível de host, métricas em nível de comando e registros de eventos disponíveis por meio do Amazon Simple Notification Service (SNS) AWS CLI e do Amazon Simple Notification Service (SNS).

Tópicos

- [Métricas para clusters baseados em nós](#)
- [Eventos para clusters baseados em nós \(Valkey e Redis OSS\)](#)

Métricas para clusters baseados em nós

Quando você cria um cluster baseado em nós, ElastiCache emite métricas em cada nível de nó, incluindo métricas em nível de host e métricas de cache.

Para obter mais informações sobre métricas no nível de host, consulte [Métricas em nível de host](#).

Para obter mais informações sobre métricas no nível de nó, consulte [Métricas para o Valkey e Redis OSS](#).

Eventos para clusters baseados em nós (Valkey e Redis OSS)

ElastiCache registra eventos relacionados aos seus clusters baseados em nós. Ao trabalhar com clusters baseados em nós, você pode visualizar seus eventos de cluster no ElastiCache console, usando o AWS CLI, ou usando o Amazon Simple Notification Service (SNS). Eventos de cluster baseados em nós não são publicados na Amazon. EventBridge

As informações do evento de cluster baseado em nós incluem a data e a hora do evento, o nome da origem e o tipo de origem do evento, bem como uma descrição do evento. Você pode recuperar facilmente os eventos do registro usando o ElastiCache console, o comando AWS CLI `describe-events` ou a ação da API `ElastiCache DescribeEvents`.

Visualizando ElastiCache eventos (Console)

O procedimento a seguir exibe eventos usando o ElastiCache console.

Para visualizar eventos usando o ElastiCache console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha Eventos.
3. Na tela Eventos, cada linha da lista representa um evento e exibe a origem do evento, o tipo de evento, o horário GMT do evento e uma descrição do evento. Usando a opção Filtro, você pode especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualizando ElastiCache eventos (AWS CLI)

Para gerar uma lista de ElastiCache eventos usando o AWS CLI, use o comando `describe-events`. Você pode usar parâmetros opcionais para controlar os tipos de eventos listados, o período de tempo dos eventos listados, o número máximo de eventos a serem listados e muito mais.

O código a seguir lista até 40 eventos de cluster baseado em nós.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

O código a seguir lista todos os eventos de cache baseado em nós nas últimas 24 horas (1.440 minutos).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Eventos de cluster baseados em nós

Esta seção contém a lista de eventos que devem ser recebidos para os clusters baseados em nós.

Os ElastiCache eventos a seguir acionam as notificações do Amazon SNS. Para obter informações sobre detalhes de eventos, consulte [Visualizando ElastiCache eventos](#).

Nome do evento	Mensagem	Description
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Um nó de cache foi adicionado ao cluster e está pronto para uso.

Nome do evento	Mensagem	Description
ElastiCache: AddCacheNodeFailed devido à insuficiência de endereços IP gratuitos	ElastiCache: AddCacheNodeFailed : <i>cluster-name</i>	Um nó de cache não pôde ser adicionado porque não há o endereços IP suficientes disponíveis.
ElastiCache: CacheClusterParametersChanged	ElastiCache: CacheClusterParametersChanged : <i>cluster-name</i>	Um ou mais parâmetros de cluster foram alterados.
ElastiCache: CacheClusterProvisioningComplete	ElastiCache: CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	O provisionamento de um cluster está concluído, e os nós de cache no cluster de cache estão prontos para uso.
ElastiCache: CacheClusterProvisioningFailed devido ao estado de rede incompatível	ElastiCache: CacheClusterProvisioningFailed : <i>cluster-name</i>	Foi feita uma tentativa de executar um novo cluster em uma nuvem privada virtual (VPC) inexistente.
ElastiCache: CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Escalabilidade para cluster de cache concluída com êxito.
ElastiCache: CacheClusterScalingFailed	ElastiCache: CacheClusterScalingFailed : <i>cluster-name</i>	A operação de expansão no cluster de cache falhou.

Nome do evento	Mensagem	Description
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	Um dos seguintes eventos ocorreu: • A lista de grupos de segurança de cache autorizados para o cluster foi modificada. • Um ou mais novos grupos EC2 de segurança foram autorizados em qualquer um dos grupos de segurança de cache associados ao cluster. • Um ou mais grupos EC2 de segurança foram revogados de qualquer um dos grupos de segurança de cache associados ao cluster.

Nome do evento	Mensagem	Description
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e começou a substituir o nó de cache.  Note A entrada de DNS para o nó de cache substituído não é alterada. Na maioria dos casos, você não precisa atualizar a lista de servidores para seus clientes quando esse evento ocorre. No entanto, algumas bibliotecas cliente de cache podem parar de usar o nó de cache mesmo após ElastiCache a substituição do nó de cache; nesse caso, o aplicativo deve atualizar a lista de servidores quando esse evento ocorrer.

Nome do evento	Mensagem	Description
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e concluiu a substituição do nó de cache.  Note A entrada de DNS para o nó de cache substituído não é alterada. Na maioria dos casos, você não precisa atualizar a lista de servidores para seus clientes quando esse evento ocorre. No entanto, algumas bibliotecas cliente de cache podem parar de usar o nó de cache mesmo após ElastiCache a substituição do nó de cache; nesse caso, o aplicativo deve atualizar a lista de servidores quando esse evento ocorrer.

Nome do evento	Mensagem	Description
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Um ou mais nós de cache foram reinicializados. Mensagem (Memcached): "Cache node %s shutdown" Em seguida, uma segunda mensagem: "Cache node %s restarted"
ElastiCache: CertificateRenewalComplete (Somente Valkey ou Redis OSS)	ElastiCache:CertificateRenewalComplete	O certificado CA da Amazon foi renovado com êxito.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	O grupo de replicação foi criado com sucesso.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	A exclusão de um cluster e todos os nós de cache associados foi concluída.
ElastiCache: FailoverComplete (Somente Valkey ou Redis OSS)	ElastiCache:FailoverComplete : <i>mycluster</i>	O failover para um nó de réplica foi bem-sucedido.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	O número de réplicas no cluster foi aumentado.

Nome do evento	Mensagem	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	O processo de adição de réplicas ao cluster começou.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Um nó no seu cluster que estava programado para substituição já não está programado para substituição.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Um nó no seu cluster previamente programado para substituição foi reprogramado para substituição durante a nova janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós (Valkey e Redis OSS).
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Um nó no seu cluster está programado para substituição durante a janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós (Valkey e Redis OSS).
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Um nó de cache foi removido do cluster.

Nome do evento	Mensagem	Description
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	A operação de expansão no grupo de replicação foi concluída com sucesso.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	A operação de expansão no grupo de replicação falhou.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Uma atualização de autoatendimento está disponível para o nó.
ElastiCache: SnapshotComplete (Somente Valkey ou Redis OSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Um snapshot de cache foi concluído com sucesso.
ElastiCache: SnapshotFailed (Somente Valkey ou Redis OSS)	SnapshotFailed : <i>cluster-name</i>	Um snapshot de cache falhou. Consulte os eventos de cache do cluster para mais uma causa detalhada. Se você descrever o snapshot, consulte DescribeSnapshots, o status será failed.

Métricas e eventos para caches e clusters do Memcached

Esta seção descreve as métricas e os eventos que você pode monitorar ao trabalhar com caches baseados em nós e com tecnologia sem servidor do Memcached.

Tópicos

- [Métricas ElastiCache sem servidor para Memcached](#)
- [Eventos de ElastiCache Serverless for Memcached](#)

Métricas ElastiCache sem servidor para Memcached

Esta seção descreve as métricas e os eventos que você pode monitorar ao trabalhar com o ElastiCache Serverless for Memcached.

O AWS/ElastiCache namespace inclui as seguintes CloudWatch métricas para seus caches sem servidor do Memcached.

Métrica	Description	Unidade
BytesUsedForCache	O número total de bytes usados pelos dados armazenados no cache.	Bytes
ElastiCacheProcessingUnits	O número total de ElastiCacheProcessingUnits (ECPUs) consumido pelas solicitações executadas em seu cache	Contagem
SuccessfulReadRequestLatency	Latência de solicitações de leitura bem-sucedidas.	Microsssegundos
SuccessfulWriteRequestLatency	Latência de solicitações de gravação bem-sucedidas	Microsssegundos
TotalCmdsCount	Contagem total de todos os comandos executados no cache	Contagem

Métrica	Description	Unidade
CurrConnections	O número de conexões cliente com o cache.	Contagem
ThrottledCmds	O número de solicitações que foram limitadas ElastiCache porque a carga de trabalho estava aumentando mais rápido do que pode ser escalada. ElastiCache	Contagem
NewConnections	O número total de conexões que foram aceitas pelo servidor durante esse período.	Contagem
CurrItems	O número de itens no cache.	Contagem
NetworkBytesIn	Total de bytes transferidos para dentro do cache	Bytes
NetworkBytesOut	Total de bytes transferidos para fora do cache	Bytes
Evictions	A contagem de chaves removidas pelo cache	Contagem
Reclaimed	O número de chaves expiradas pelo cache	Contagem

Métricas no nível de comando

ElastiCache também emite as seguintes métricas de nível de comando do Memcached

Métrica	Description	Unidade
CmdGet	O número de comandos get recebidos pelo cache.	Contagem

Métrica	Description	Unidade
CmdSet	O número de comandos set recebidos pelo cache.	Contagem
CmdTouch	O número de comandos touch recebidos pelo cache.	Contagem
GetHits	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
GetMisses	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
IncrHits	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
IncrMisses	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
DecrHits	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
DecrMisses	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem

Métrica	Description	Unidade
DeleteHits	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
DeleteMisses	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
TouchHits	O número de chaves que foram tocadas e receberam um novo horário de expiração.	Contagem
TouchMisses	O número de chaves que foram tocadas e não foram encontradas.	Contagem
CasHits	O número de solicitações cas recebidas pelo cache em que a chave solicitada foi encontrada e o valor de cas correspondia.	Contagem
CasMisses	O número de solicitações cas recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
CasBadval	O número de solicitações cas recebidas pelo cache em que o valor de cas não correspondeu ao valor de cas armazenado.	Contagem

Métrica	Description	Unidade
CmdFlush	O número de comandos flush recebidos pelo cache.	Contagem

Eventos de ElastiCache Serverless for Memcached

ElastiCache registra eventos relacionados ao seu cache sem servidor. Essas informações incluem a data e a hora do evento, o nome da origem e o tipo de origem do evento, bem como uma descrição do evento. Você pode facilmente recuperar eventos do registro usando o ElastiCache console, o comando AWS CLI `describe-events` ou a ação da API `ElastiCache DescribeEvents`.

Você pode escolher monitorar, ingerir, transformar e agir em ElastiCache eventos usando a Amazon EventBridge. Saiba mais no [guia de primeiros EventBridge passos](#) da Amazon.

Visualizando ElastiCache eventos (console)

Para visualizar eventos usando o ElastiCache console:

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha Eventos.
3. Na tela Eventos, cada linha da lista representa um evento e exibe a origem do evento, o tipo de evento, o horário GMT do evento e uma descrição do evento. Usando a opção Filtro, você pode especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualizando ElastiCache eventos (AWS CLI)

Para gerar uma lista de ElastiCache eventos usando o AWS CLI, use o comando `describe-events`. Você pode usar parâmetros opcionais para controlar os tipos de eventos listados, o período de tempo dos eventos listados, o número máximo de eventos a serem listados e muito mais.

O código a seguir lista até 40 eventos de cache sem servidor.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

O código a seguir lista todos os eventos de cache sem servidor nas últimas 24 horas (1.440 minutos).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Eventos sem servidor

Esta seção documenta os diferentes tipos de eventos que você pode receber para os caches sem servidor.

Eventos de criação de cache sem servidor

Detail-Type	Description	Unidade	Fonte	Mensagem
Cache criado	Fio de cache	criação	serverless-cache	O cache <cache-name> foi criado e está pronto para ser usado.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Endereços IP livres insuficientes para criar um endpoint da VPC.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Sub-redes inválidas fornecidas na solicitação.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache

Detail-Type	Description	Unidade	Fonte	Mensagem
				<cache-name>. Limite de cota atingido para criação de um endpoint da VPC.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Você não tem permissões para criar um endpoint da VPC.

Eventos de atualização de cache com tecnologia sem servidor (Memcached)

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	SecurityGroups atualizado para o cache<cache-name>.
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	Tags atualizadas para o cache <cache-name>.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do cache. SecurityGroups falha na atualização.

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do cache. SecurityGroups a atualização falhou devido a permissões insuficientes.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do cache. SecurityGroups a atualização falhou porque SecurityGroups são inválidos.

Eventos de exclusão do cache com tecnologia sem servidor (Memcached)

Detail-Type	Lista de recursos	Categoria	Fonte	Mensagem
Cache excluído	Fio de cache	exclusão	serverless-cache	O cache <cache-name> foi excluído.

Eventos de limite de uso do cache com tecnologia sem servidor (Memcached)

Detail-Type	Description	Unidade	Fonte	Mensagem
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	Limites atualizados para o cache <cache-name>.

Detail-Type	Description	Unidade	Fonte	Mensagem
Falha na atualização do cache	Fio de cache	falha	serverless-cache	Falha na atualização de um limite para o cache <cache-name> porque o cache foi excluído.
Falha na atualização do cache	Fio de cache	falha	serverless-cache	Falha na atualização de um limite no cache <cache-name> por causa da configuração inválida.

Eventos de snapshot de cache com tecnologia sem servidor (Memcached)

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Snapshot criado	Fio de cache Fio de snapshot	criação	serverless-cache-snapshot	Snapshot <snapshot-name> criado para cache <cache-name>.
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	Falha na criação do snapshot para o cache <cache-name>. Falha na criação do snapshot <snapshot-name> com a chave gerenciad

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
				a pelo cliente <key-id> <reason>. Mensagens de motivo da falha: • porque a chave gerenciada pelo cliente está desabilitada • porque não foi possível encontrar a chave gerenciada pelo cliente • porque a solicitação atingiu o tempo limite

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	Falha na criação do snapshot para o cache <cache-name>. Houve falha na criação do snapshot <snapshot-name> por causa de <reason>. Motivo padrão: por causa de um erro interno
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o instantâneo para o bucket %s porque ElastiCache não tem permissões para o bucket.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque já existe um objeto com o mesmo nome no bucket.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o instantâneo para o bucket '%s' porque o ID da conta do proprietário do bucket foi alterado.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque o bucket do S3 não está acessível.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque o bucket não está acessível.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' porque o bucket não existe.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s' com a chave gerenciada pelo cliente do snapshot de origem %s <reason>.
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket '%s'.

Detail-Type	Resources-list	Categoria	Fonte	Mensagem
Falha na cópia do snapshot	Snapshot arn-1 Snapshot arn-2	falha	serverless-cache-snapshot	Falha na cópia do snapshot <snapshot-name>. Não foi possível copiar o snapshot '%s' para o snapshot '%s' com a chave gerenciada pelo cliente do snapshot de origem <key-id> <reason-name>.
Falha na cópia do snapshot	Snapshot arn-1 Snapshot arn-2	falha	serverless-cache-snapshot	Falha na cópia do snapshot <snapshot-name>. Não foi possível copiar o snapshot '%s' para o snapshot '%s' com a chave gerenciada pelo cliente do instantâneo de destino '%s' '%s'.

Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail

O Amazon ElastiCache é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, por uma função ou por um serviço da AWS no Amazon ElastiCache. O CloudTrail captura todas as chamadas de API para o Amazon ElastiCache como eventos, inclusive as chamadas do console do Amazon ElastiCache e de chamadas do código

para operações da API do Amazon ElastiCache. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Amazon ElastiCache. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos. Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o Amazon ElastiCache, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail CloudTrail](#)).

Informações sobre o Amazon ElastiCache no CloudTrail

O CloudTrail é habilitado em sua conta AWS ao criá-la. Quando ocorre uma atividade no Amazon ElastiCache, essa atividade é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS no Event history (Histórico de eventos). É possível visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro de eventos em andamento na sua conta da AWS, incluindo eventos do Amazon ElastiCache, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, ao criar uma trilha no console, ela é aplicada a todas as regiões da . A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros AWS serviços para melhor analisar e agir de acordo com dados coletados do evento nos logs CloudTrail. Para obter mais informações, consulte:

- [Visão Geral para Criar uma Trilha](#)
- [Serviços e integrações com suporte no CloudTrail](#)
- [Configurando Notificações Amazon SNS para CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#) e [receber arquivos de log do CloudTrail de várias contas](#)

Todas as ações do Amazon ElastiCache são registradas pelo CloudTrail e estão documentadas na [Referência da API do ElastiCache](#). Por exemplo, as chamadas às ações CreateCacheCluster, DescribeCacheCluster e ModifyCacheCluster geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário-raiz ou usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para mais informações, consulte [Elemento userIdentity CloudTrail](#).

Noções básicas sobre entradas de arquivo de log do Amazon ElastiCache

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket Amazon S3 especificado. Os arquivos de log CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `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"
  }
}

```

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `DescribeCacheCluster`. Observe que, para todas as chamadas `Describe` do Amazon ElastiCache (`Describe*`), a seção `ResponseElements` é removida e aparece como `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"
}

```

O exemplo a seguir mostra uma entrada de log do CloudTrail que registra uma ação `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"
  }
}
```

Monitoramento de eventos do ElastiCache pelo Amazon SNS

Quando ocorrem eventos significativos para um cluster, o ElastiCache envia uma notificação para um tópico específico do Amazon SNS. Exemplos incluem uma falha ao adicionar um nó, êxito ao

adicionar um nó, a modificação de um grupo de segurança, e outros. Ao monitorar eventos chave, você pode se manter informado sobre o atual estado dos seus clusters e, dependendo do evento, poderá executar uma ação corretiva.

Tópicos

- [Gerenciando ElastiCache notificações do Amazon SNS](#)
- [Visualizando ElastiCache eventos](#)
- [Notificações de eventos e o Amazon SNS](#)

Gerenciando ElastiCache notificações do Amazon SNS

Você pode configurar o ElastiCache envio de notificações para eventos importantes do cluster usando o Amazon Simple Notification Service (Amazon SNS). Nestes exemplos, você configurará um cluster com o nome de recurso da Amazon (ARN) de um tópico do Amazon SNS para receber notificações.

Note

- Esse tópico pressupõe que você tenha se cadastrado no Amazon SNS e configurado e assinado um tópico do Amazon SNS. Para obter informações sobre como fazer isso, consulte o [Guia do desenvolvedor do Amazon Simple Notification Service](#).
- Por padrão, API `modify-replication-group` afeta todos os grupos em uma região e não apenas o grupo especificado atualmente. Se você quiser configurar um grupo específico em uma região de forma diferente dos outros grupos, você pode usar a opção `--notification-topic-arn` para criar um tópico separado para esse grupo.

Adição de um tópico do Amazon SNS

As seções a seguir mostram como adicionar um tópico do Amazon SNS usando o AWS console AWS CLI, o ou a ElastiCache API.

Adição de um tópico do Amazon SNS (console)

O procedimento a seguir mostra como adicionar um tópico do Amazon SNS para um cluster. Ao usar o Valkey ou o Redis OSS para adicionar um tópico do Amazon SNS para um grupo de replicação

na etapa 2, em vez de escolher um cluster, escolha um grupo de replicação. Em seguida, siga as mesmas etapas restantes.

 Note

Esse processo também pode ser usado para modificar o tópico do Amazon SNS.

Para adicionar ou modificar um tópico do Amazon SNS para um cluster (console)

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Em Clusters, escolha o cluster para o qual deseja adicionar ou modificar um ARN de tópico do Amazon SNS.
3. Escolha Modificar.
4. Em Modificar cluster em Tópico para notificação do SNS, escolha o tópico SNS que você deseja adicionar ou escolha Entrada manual de ARN e insira o ARN do tópico do Amazon SNS.
5. Escolha Modificar.

Adição de um tópico do Amazon SNS (AWS CLI)

Para adicionar ou modificar um tópico do Amazon SNS para um cluster, use o AWS CLI comando. `modify-cache-cluster`

O seguinte exemplo de código adiciona um ARN de tópico do Amazon SNS a my-cluster.

Para 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
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Para obter mais informações, consulte [modify-cache-cluster](#).

Adicionar um tópico do Amazon SNS (API) ElastiCache

Para adicionar ou modificar um tópico do Amazon SNS para um cluster, chame a ação `ModifyCacheCluster` com os seguintes parâmetros:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
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>
```

Para obter mais informações, consulte [ModifyCacheCluster](#).

Habilitação e desabilitação de notificações do Amazon SNS

Você pode ativar ou desativar notificações para um cluster. Os procedimentos a seguir mostram como desativar notificações do Amazon SNS.

Habilitação e desabilitação de notificações do Amazon SNS (console)

Para desativar as notificações do Amazon SNS usando o Console de gerenciamento da AWS

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista dos clusters que estão executando o Memcached, no painel de navegação, escolha Memcached.

Para ver uma lista dos clusters que estão executando o Valkey ou o Redis OSS, no painel de navegação, selecione Valkey ou Redis OSS.

3. Escolha a caixa à esquerda do nome do cluster para o qual você deseja modificar notificações.
4. Escolha Modificar.
5. Em Modify Cluster, em Topic for SNS Notification, escolha Disable Notifications.
6. Escolha Modificar.

Habilitação e desabilitação de notificações do Amazon SNS (AWS CLI)

Para desabilitar notificações do Amazon SNS, use o comando `modify-cache-cluster` com os seguintes parâmetros:

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-status inactive
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-status inactive
```

Note

Quando o cluster pertence a um grupo de replicação, você deve usar o comando `modify-replication-group` da CLI para habilitar ou desabilitar as notificações do SNS.

Ativando e desativando as notificações ElastiCache do Amazon SNS (API)

Para desabilitar notificações do Amazon SNS, chame a ação `ModifyCacheCluster` com os seguintes parâmetros:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Essa chamada retorna uma saída semelhante à seguinte:

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=false  
&CacheClusterId=my-cluster  
&NotificationTopicStatus=inactive  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Visualizando ElastiCache eventos

ElastiCache registra eventos relacionados às suas instâncias de cluster, grupos de segurança e grupos de parâmetros. Essas informações incluem a data e a hora do evento, o nome da origem e o tipo de origem do evento, bem como uma descrição do evento. Você pode recuperar facilmente os eventos do registro usando o ElastiCache console, o AWS CLI `describe-events` comando ou a ação `DescribeEvents` da ElastiCache API.

Os procedimentos a seguir mostram como visualizar todos os ElastiCache eventos das últimas 24 horas (1440 minutos).

Visualizando ElastiCache eventos (Console)

O procedimento a seguir exibe eventos usando o ElastiCache console.

Para visualizar eventos usando o ElastiCache console

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha Eventos.

Na tela Eventos, cada linha da lista representa um evento e exibe a origem do evento, o tipo de evento (cache-cluster, cache-parameter-group, cache-security-group, ou cache-subnet-group), a hora GMT do evento e uma descrição do evento.

Usando a opção Filtro, você pode especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualizando ElastiCache eventos (AWS CLI)

Para gerar uma lista de ElastiCache eventos usando o AWS CLI, use o comando `describe-events`. Você pode usar parâmetros opcionais para controlar os tipos de eventos listados, o período de tempo dos eventos listados, o número máximo de eventos a serem listados e muito mais.

O código a seguir lista até 40 eventos de cluster.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

O código a seguir lista todos os eventos nas últimas 24 horas (1440 minutos).


```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

A saída do comando `describe-events` é semelhante a esta.

```
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 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 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 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"
    }
```

```
    }  
  ]  
}
```

Para obter mais informações, como os parâmetros disponíveis e os valores de parâmetros permitidos, consulte [describe-events](#).

Visualização de ElastiCache eventos (ElastiCache API)

Para gerar uma lista de ElastiCache eventos usando a ElastiCache API, use a `DescribeEvents` ação. Você pode usar parâmetros opcionais para controlar os tipos de eventos listados, o período de tempo dos eventos listados, o número máximo de eventos a serem listados e muito mais.

O código a seguir lista os 40 eventos de cluster de cache mais recentes.

```
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>
```

O código a seguir lista os eventos de cluster de cache nas últimas 24 horas (1440 minutos).

```
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>
```

As ações acima devem produzir uma saída semelhante à seguinte.

```
<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>
```

Para obter mais informações, como os parâmetros disponíveis e os valores de parâmetros permitidos, consulte [DescribeEvents](#).

Notificações de eventos e o Amazon SNS

ElastiCache pode publicar mensagens usando o Amazon Simple Notification Service (SNS) quando eventos significativos acontecem em um cluster. Esse recurso pode ser usado para atualizar as listas de servidores em máquinas clientes conectadas a endpoints de nó de cache individuais de um cluster de cache.

Note

Para obter mais informações sobre o Amazon Simple Notification Service (SNS), incluindo informações sobre preços e links para a documentação do Amazon SNS, consulte a [página do produto Amazon SNS](#).

As notificações são publicadas em um tópico do Amazon SNS especificado. Os seguintes são requisitos para notificações:

- Somente um tópico pode ser configurado para ElastiCache notificações.
- A AWS conta proprietária do tópico do Amazon SNS deve ser a mesma conta que possui o cluster no qual as notificações estão habilitadas.
- O tópico do Amazon SNS no qual você está publicando não pode ser criptografado.

Note

É possível anexar um tópico criptografado (em repouso) do Amazon SNS ao cluster. No entanto, o status do tópico no ElastiCache console será exibido como inativo, o que efetivamente dissocia o tópico do cluster ao enviar mensagens para ElastiCache o tópico.

- O tópico do Amazon SNS precisa estar na mesma região do cluster. ElastiCache

ElastiCache Eventos

Os ElastiCache eventos a seguir acionam as notificações do Amazon SNS. Para obter informações sobre detalhes de eventos, consulte [Visualizando ElastiCache eventos](#).

Nome do evento	Mensagem	Description
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Um nó de cache foi adicionado ao cluster e está pronto para uso.
ElastiCache: AddCacheNodeFailed devido à insuficiência de endereços IP gratuitos	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Um nó de cache não pôde ser adicionado porque não há o endereços IP suficientes disponíveis.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Um ou mais parâmetros de cluster foram alterados.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	O provisionamento de um cluster está concluído, e os nós de cache no cluster de cache estão prontos para uso.
ElastiCache: CacheClusterProvisioningFailed devido ao estado de rede incompatível	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Foi feita uma tentativa de executar um novo cluster em uma nuvem privada virtual (VPC) inexistente.

Nome do evento	Mensagem	Description
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Escalabilidade para cluster de cache concluída com êxito.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	A operação de expansão no cluster de cache falhou.
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	Um dos seguintes eventos ocorreu: • A lista de grupos de segurança de cache autorizados para o cluster foi modificada. • Um ou mais novos grupos EC2 de segurança foram autorizados em qualquer um dos grupos de segurança de cache associados ao cluster. • Um ou mais grupos EC2 de segurança foram revogados de qualquer um dos grupos de segurança de cache associados ao cluster.

Nome do evento	Mensagem	Description
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e começou a substituir o nó de cache.  Note A entrada de DNS para o nó de cache substituído não é alterada. Na maioria dos casos, você não precisa atualizar a lista de servidores para seus clientes quando esse evento ocorre. No entanto, algumas bibliotecas cliente de cache podem parar de usar o nó de cache mesmo após ElastiCache a substituição do nó de cache; nesse caso, o aplicativo deve atualizar a lista de servidores quando esse evento ocorrer.

Nome do evento	Mensagem	Description
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e concluiu a substituição do nó de cache.  Note A entrada de DNS para o nó de cache substituído não é alterada. Na maioria dos casos, você não precisa atualizar a lista de servidores para seus clientes quando esse evento ocorre. No entanto, algumas bibliotecas cliente de cache podem parar de usar o nó de cache mesmo após ElastiCache a substituição do nó de cache; nesse caso, o aplicativo deve atualizar a lista de servidores quando esse evento ocorrer.

Nome do evento	Mensagem	Description
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Um ou mais nós de cache foram reinicializados. Mensagem (Memcached): "Cache node %s shutdown" Em seguida, uma segunda mensagem: "Cache node %s restarted"
ElastiCache: CertificateRenewalComplete (Somente Valkey ou Redis OSS)	ElastiCache:CertificateRenewalComplete	O certificado CA da Amazon foi renovado com êxito.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	O grupo de replicação foi criado com sucesso.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	A exclusão de um cluster e todos os nós de cache associados foi concluída.
ElastiCache: FailoverComplete (Somente Valkey ou Redis OSS)	ElastiCache:FailoverComplete : <i>mycluster</i>	O failover para um nó de réplica foi bem-sucedido.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	O número de réplicas no cluster foi aumentado.

Nome do evento	Mensagem	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	O processo de adição de réplicas ao cluster começou.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Um nó no seu cluster que estava programado para substituição já não está programado para substituição.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Um nó no seu cluster previamente programado para substituição foi reprogramado para substituição durante a nova janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós (Valkey e Redis OSS).
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Um nó no seu cluster está programado para substituição durante a janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós (Valkey e Redis OSS).
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Um nó de cache foi removido do cluster.

Nome do evento	Mensagem	Description
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	A operação de expansão no grupo de replicação foi concluída com sucesso.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	A operação de expansão no grupo de replicação falhou.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Uma atualização de autoatendimento está disponível para o nó.
ElastiCache: SnapshotComplete (Somente Valkey ou Redis OSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Um snapshot de cache foi concluído com sucesso.
ElastiCache: SnapshotFailed (Somente Valkey ou Redis OSS)	SnapshotFailed : <i>cluster-name</i>	Um snapshot de cache falhou. Consulte os eventos de cache do cluster para mais uma causa detalhada. Se você descrever o snapshot, consulte DescribeSnapshots, o status será failed.

Tópicos relacionados

- [Visualizando ElastiCache eventos](#)

Entrega de logs

Note

O log lento é compatível com o Valkey 7.x e versões posteriores, e clusters e grupos de replicação do Redis OSS que usam o mecanismo versão 6.0 em diante.

O log do mecanismo é compatível com clusters e grupos de replicação do Valkey 7.x e superiores e do Redis OSS que usam o mecanismo versão 6.2 em diante.

A entrega de logs permite que você transmita [SLOWLOG](#) ou log do mecanismo para um de dois destinos:

- Amazon Data Firehose
- CloudWatch Registros da Amazon

Você ativa e configura a entrega de registros ao criar ou modificar um cluster usando ElastiCache APIs. Cada entrada de log será entregue ao destino especificado em um dos dois formatos: JSON ou TEXT.

Um número fixo de entradas de log lento são recuperados periodicamente do mecanismo. Dependendo do valor especificado para o parâmetro do mecanismo `slowlog-max-len`, é possível que entradas de log lentas adicionais não sejam entregues no destino.

Você pode optar por alterar as configurações de entrega ou desativar a entrega de registros a qualquer momento usando o AWS console ou uma das modificações APIs, [modify-cache-cluster](#) ou [modify-replication-group](#).

Você deve definir o parâmetro `apply-immediately` para todas as modificações de entrega de log.

Note

As cobranças do Amazon CloudWatch Logs se aplicam quando a entrega de registros está ativada, mesmo quando os registros são entregues diretamente ao Amazon Data Firehose. Para obter mais informações, consulte a seção Vended Logs na [Amazon CloudWatch Pricing](#).

Conteúdo de uma entrada de log lento

O log lento contém as seguintes informações:

- **CacheClusterId**— O ID do cluster
- **CacheNodeId**— O ID do nó de cache
- **Id**: um identificador progressivo exclusivo para cada entrada de log lento
- **Timestamp**: O timestamp do Unix no qual o comando registrado foi processado
- **Duration (Duração)**: a quantidade de tempo necessário para sua execução, em microssegundos
- **Command (Comando)**: o comando usado pelo cliente. Por exemplo, `set foo bar` onde `foo` está a chave e `bar` o valor. ElastiCache substitui o nome e o valor reais da chave por `(2 more arguments)` para evitar a exposição de dados confidenciais.
- **ClientAddress**— Endereço IP e porta do cliente
- **ClientName**— Nome do cliente, se definido por meio do `CLIENT SETNAME` comando

Conteúdo de uma entrada de log do mecanismo

O ElastiCache Engine Log contém as seguintes informações:

- **CacheClusterId**— O ID do cluster
- **CacheNodeId**— O ID do nó de cache
- **Nível de registro** — **LogLevel** pode ser um dos seguintes: `VERBOSE("-")`, `NOTICE("*")`, `WARNING("#")`.
- **Time (Hora)** - O horário UTC da mensagem registrada. A hora está no seguinte formato: `"DD MMM YYYY hh:mm:ss.ms UTC"`
- **Role (Função)** - Função do nó de onde o log é emitido. Pode ser uma das seguintes opções: `"M"` para Primária, `"S"` para réplica, `"C"` para escritor que trabalha com o processo infantil RDB/AOF ou `"X"` para sentinela.
- **Mensagem**: mensagem de log do mecanismo.

Permissões para configurar os logs

Você precisa incluir as seguintes permissões do IAM na sua user/role política do IAM:

- `logs:CreateLogDelivery`

- logs:UpdateLogDelivery
- logs>DeleteLogDelivery
- logs:GetLogDelivery
- logs:ListLogDeliveries

Para obter mais informações, consulte [Visão geral do gerenciamento de acesso: permissões e políticas](#).

Especificações de tipo de log e formato de log

Log lento

O log lento suporta JSON e TEXT

O exemplo a seguir mostra um exemplo de formato JSON.

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

O exemplo a seguir mostra um exemplo de formato TEXT.

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Log do mecanismo

O log do mecanismo é compatível com JSON e TEXT

O exemplo a seguir mostra um exemplo de formato JSON.

```
{
  "CacheClusterId": "xxxxxxxxzy-engine-log-test",
```

```
"CacheNodeId": "0001",  
"LogLevel": "VERBOSE",  
"Role": "M",  
"Time": "12 Nov 2020 01:28:57.994 UTC",  
"Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.  
Check back later"  
}
```

O exemplo a seguir mostra um exemplo de formato TEXT.

```
xxxxxxxxxxxxzy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua  
script detected that is still in execution after 1000 microseconds.
```

ElastiCache destinos de registro

Esta seção descreve os destinos de registro que você pode escolher para seus ElastiCache registros. Cada seção fornece orientações para configurar os logs para o tipo de destino e informações sobre qualquer comportamento específico para o tipo de destino. Depois de configurar seu destino de registro, você pode fornecer suas especificações para a configuração de ElastiCache registro para começar a registrá-lo.

Tópicos

- [CloudWatch Registros da Amazon](#)
- [Amazon Data Firehose](#)

CloudWatch Registros da Amazon

- Você especifica um grupo de CloudWatch registros de registros em que os registros serão entregues.
- Os logs de vários clusters do Valkey ou Redis OSS e grupos de replicação podem ser entregues ao mesmo grupo de logs.
- Um novo fluxo de log será criado para cada nó dentro de um cluster ou grupo de replicação e os logs serão entregues aos respectivos fluxos de log. O nome do fluxo de log usará o seguinte formato: `elasticache/${engine-name}/${cache-cluster-id}/${cache-node-id}/${log-type}`

Permissões para publicar registros no CloudWatch Logs

Você precisa ter as seguintes configurações de permissões para configurar o ElastiCache envio de registros para um grupo de CloudWatch registros de registros:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingCWL",
      "Action": [
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Para obter mais informações, consulte [Registros enviados para o CloudWatch Logs](#).

Amazon Data Firehose

- Você especifica um fluxo de entrega do Firehose no qual os logs serão entregues.
- Os logs de vários grupos de replicação e clusters do Valkey ou Redis OSS podem ser entregues ao mesmo fluxo de entrega.
- Os logs de cada nó dentro de um cluster ou grupo de replicação serão entregues ao mesmo fluxo de entrega. Você pode distinguir mensagens de log de diferentes nós de cache com base no `cache-cluster-id` e no `cache-node-id` incluído em cada mensagem de log.
- No momento, a entrega de logs para o Firehose não está disponível na região Asia Pacific (Osaka).

Permissões para publicar logs no Firehose

Você deve ter as seguintes permissões para configurar o envio de registros ElastiCache para um stream de entrega do Amazon Kinesis Data Firehose.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingFHSLR",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
```

```
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Sid": "ElastiCacheLoggingFH",
        "Action": [
            "firehose:TagDeliveryStream"
        ],
        "Resource": "arn:aws:iam::*:role/*",
        "Effect": "Allow"
    }
]
```

Especificação da entrega de logs usando o console

Ao usar o Console de gerenciamento da AWS, você pode criar um cluster do Valkey ou Redis OSS (modo cluster desabilitado) seguindo as etapas em [Criação de um cluster do Valkey \(modo cluster desabilitado\) \(console\)](#) ou criar um cluster do Valkey ou Redis OSS (modo cluster habilitado) usando as etapas em [Criação de um cluster do Valkey ou Redis OSS \(modo cluster habilitado\) \(console\)](#). Em ambos os casos, você configura a entrega de logs fazendo o seguinte:

1. Em Configurações avançadas, escolha Logs e, em seguida, Logs lentos ou Logs de mecanismos.
2. Em Log format (Formato do log), escolha Text (Texto) ou JSON.
3. Em Destination Type (Tipo de destino), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.
4. Em Log destination (Destino do log), escolha Create new (Criar) e insira o nome do bucket do Amazon S3, o nome do grupo de logs do CloudWatchLogs ou o nome do fluxo do Kinesis Data Firehose, ou escolha Select existing (Selecionar existente) e, em seguida, escolha o nome do grupo de logs do CloudWatchLogs ou o nome do fluxo do Kinesis Data Firehose,

Ao modificar um cluster:

Você pode optar por habilitar/desabilitar a entrega de logs ou alterar o tipo de destino, o formato ou o destino:

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Clusters do Valkey ou Clusters do Redis OSS.
3. Na lista de clusters, escolha o cluster que você deseja modificar. Selecione o Cluster name (Nome do cluster), e não a caixa de seleção ao lado dele.
4. Na página Cluster name (Nome do cluster), escolha a guia Logs.
5. Para habilitar/desabilitar logs lentos, escolha Enable slow logs (Habilitar logs lentos) ou Disable slow logs (Desabilitar logs lentos).
6. Para ativar/desativar os logs do mecanismo, escolha Enable engine logs (Ativar logs do mecanismo) ou Disable engine logs (Desativar logs do mecanismo).
7. Para alterar sua configuração, escolha Modify slow logs (Modificar logs lentos) ou Modify engine logs (Modificar logs do mecanismo):
 - Em Destination Type (Tipo de destino), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.
 - Em Log destination (Destino do log), escolha Create new (Criar novo) e insira o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose. Ou escolha Select existing (Selecionar existente) e escolha o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose.

Especificação da entrega de log usando a AWS CLI

Log lento

Crie um grupo de replicação com entrega de logs lentos para o CloudWatch Logs.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id test-slow-log \  
  --replication-group-description test-slow-log \  
  --engine redis \  
  --cache-node-type cache.r5.large \  
  --num-cache-clusters 2 \  
  --log-delivery-configurations '{  
    "LogType":"slow-log",  
    "DestinationType":"cloudwatch-logs",
```

```
"DestinationDetails":{
  "CloudWatchLogsDetails":{
    "LogGroup":"my-log-group"
  }
},
"LogFormat":"json"
}'
```

Para Windows:

```
aws elasticache create-replication-group ^
--replication-group-id test-slow-log ^
--replication-group-description test-slow-log ^
--engine redis ^
--cache-node-type cache.r5.large ^
--num-cache-clusters 2 ^
--log-delivery-configurations '{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{
      "LogGroup":"my-log-group"
    }
  },
  "LogFormat":"json"
}'
```

Modifique um grupo de replicação para entregar logs lentos para o CloudWatch Logs

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \
--replication-group-id test-slow-log \
--apply-immediately \
--log-delivery-configurations '
{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{

      "LogGroup":"my-log-group"
    }
  }
}
```

```
    },  
    "LogFormat":"json"  
  }'
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {  
    "LogType":"slow-log",  
    "DestinationType":"cloudwatch-logs",  
    "DestinationDetails":{  
      "CloudWatchLogsDetails":{  
        "LogGroup":"my-log-group"  
      }  
    },  
    "LogFormat":"json"  
  }'
```

Modifique um grupo de replicação para desabilitar a entrega de logs lentos

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
  {  
    "LogType":"slow-log",  
    "Enabled":false  
  }'
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {  
    "LogType":"slow-log",
```

```
"Enabled":false
}'
```

Log do mecanismo

Crie um grupo de replicação com entrega de logs do mecanismo para o CloudWatch Logs.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \
  --replication-group-id test-slow-log \
  --replication-group-description test-slow-log \
  --engine redis \
  --cache-node-type cache.r5.large \
  --num-cache-clusters 2 \
  --log-delivery-configurations '{
    "LogType":"engine-log",
    "DestinationType":"cloudwatch-logs",
    "DestinationDetails":{"
      "CloudWatchLogsDetails":{"
        "LogGroup":"my-log-group"
      }
    },
    "LogFormat":"json"
  }'
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id test-slow-log ^
  --replication-group-description test-slow-log ^
  --engine redis ^
  --cache-node-type cache.r5.large ^
  --num-cache-clusters 2 ^
  --log-delivery-configurations '{
    "LogType":"engine-log",
    "DestinationType":"cloudwatch-logs",
    "DestinationDetails":{"
      "CloudWatchLogsDetails":{"
        "LogGroup":"my-log-group"
      }
    },
    "LogFormat":"json"
  }'
```

```
"LogFormat":"json"
}'
```

Modifique um grupo de replicação para entregar logs do mecanismo para o Firehose

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \
  --replication-group-id test-slow-log \
  --apply-immediately \
  --log-delivery-configurations '
{
  "LogType":"engine-log",
  "DestinationType":"kinesis-firehose",
  "DestinationDetails":{
    "KinesisFirehoseDetails":{
      "DeliveryStream":"test"
    }
  },
  "LogFormat":"json"
}'
```

Para Windows:

```
aws elasticache modify-replication-group ^
  --replication-group-id test-slow-log ^
  --apply-immediately ^
  --log-delivery-configurations '
{
  "LogType":"engine-log",
  "DestinationType":"kinesis-firehose",
  "DestinationDetails":{
    "KinesisFirehoseDetails":{
      "DeliveryStream":"test"
    }
  },
  "LogFormat":"json"
}'
```

Modifique um grupo de replicação para alternar para o formato de mecanismo

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "LogFormat":"json"  
  }'
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "LogFormat":"json"  
  }'
```

Modifique um grupo de replicação para desabilitar a entrega de logs de mecanismo

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "Enabled":false  
  }'
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "Enabled":false  
  }'
```



```
}'
```

Monitorando o uso com CloudWatch métricas

ElastiCache fornece métricas que permitem monitorar seus clusters. Você pode acessar essas métricas por meio de CloudWatch. Para obter mais informações sobre CloudWatch, consulte a [CloudWatch documentação](#).

ElastiCache fornece métricas em nível de host (por exemplo, uso da CPU) e métricas específicas do software do mecanismo de cache (por exemplo, entradas e perdas de cache). Essas métricas são medidas e publicadas para cada nó de cache em intervalos de 60 segundos.

Important

Você deve considerar a configuração de CloudWatch alarmes em determinadas métricas principais, para ser notificado se o desempenho do seu cluster começar a diminuir. Para obter mais informações, consulte [Que métricas devo monitorar?](#) neste guia.

Tópicos

- [Métricas em nível de host](#)
- [Métricas para o Valkey e Redis OSS](#)
- [Métricas para o Memcached](#)
- [Que métricas devo monitorar?](#)
- [Escolher estatísticas e períodos de métricas](#)
- [Monitorando métricas de CloudWatch clusters e nós](#)

Métricas em nível de host

O namespace AWS/ElastiCache inclui as seguintes métricas no nível de host para nós de cache individuais. Essas métricas são medidas e publicadas para cada nó de cache em intervalos de 60 segundos.

Consulte também

- [Métricas para o Valkey e Redis OSS](#)

Métrica	Descrição	Unidade
CPUUtilization	A percentagem de utilização da CPU para o host inteiro. Como o Valkey e o Redis OSS são de thread único, recomendamos que você monitore a métrica EngineCPUUtilization , para nós com quatro ou mais vCPUs.	Percentual
CPUCreditBalance	O número de créditos ganhos de CPU que uma instância acumulou desde que foi executada ou iniciada. Para a T2 Padrão, o CPUCreditBalance também inclui o número de créditos de execução que foram acumulados. Os créditos são acumulados no saldo de créditos após terem sido ganhos e são removidos do saldo de créditos quando são gastos. O saldo de crédito tem um limite máximo, determinado pelo tamanho da instância. Depois que o limite for atingido, todos os novos créditos ganhos serão descartados. Para a T2 Padrão, os créditos de execução não são contabilizados para o limite. Os créditos do CPUCreditBalance são disponibilizados para que a instância gaste e apresente intermitência com uma utilização de CPU acima da linha de base. As métricas de crédito de CPU estão disponíveis a uma frequência de apenas 5 minutos. Essa métrica não está disponível para instâncias expansíveis T2.	Créditos (minutos de vCPU)

Métrica	Descrição	Unidade
CPUCreditUsage	O número de créditos de CPU gastos pela instância por utilização de CPU. Um crédito de CPU equivale a um vCPU em execução em 100% de utilização por um minuto ou a uma combinação equivalente de vCPUs, utilização e tempo (por exemplo, um vCPU em execução a 50% de utilização por dois minutos ou dois vCPUs em execução a 25% de utilização por dois minutos). As métricas de crédito de CPU estão disponíveis a uma frequência de apenas 5 minutos. Se você especificar um período de mais cinco minutos, use a estatística Sum (Soma) em vez da estatística Average (Média). Essa métrica não está disponível para instâncias expansíveis T2.	Créditos (minutos de vCPU)
FreeableMemory	A quantidade de memória livre disponível no host. Ela é derivada da RAM, dos buffers e do cache que o sistema operacional relata como passíveis de liberação.	Bytes
NetworkBytesIn	O número de bytes que o host leu da rede.	Bytes
NetworkBytesOut	O número de bytes enviados em todas as interfaces de rede pela instância.	Bytes
NetworkPacketsIn	O número de pacotes recebidos em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de entrada em termos do número de pacotes em uma única instância.	Contagem

Métrica	Descrição	Unidade
NetworkPacketsOut	O número de pacotes enviados em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de saída em termos do número de pacotes em uma única instância.	Contagem
NetworkBandwidthInAllowanceExceeded	Número de pacotes na fila ou descartados porque a largura de banda agregada de entrada excedeu o máximo para a instância.	Contagem
NetworkConntrackAllowanceExceeded	Número de pacotes descartados porque o monitoramento da conexão excedeu o máximo para a instância e não foi possível estabelecer novas conexões. Isso pode resultar em perda de pacotes para tráfego indo para a instância ou vindo da instância	Contagem
NetworkBandwidthOutAllowanceExceeded	Número de pacotes na fila ou descartados porque a largura de banda agregada de saída excedeu o máximo para a instância.	Contagem
NetworkPacketsPerSecondAllowanceExceeded	O número de pacotes na fila ou descartados porque o valor bidirecional de pacotes por segundo excedeu o máximo para a instância.	Contagem
NetworkMaxBytesIn	A intermitência máxima por segundo de bytes recebidos em cada minuto.	Bytes
NetworkMaxBytesOut	A intermitência máxima por segundo de bytes transmitidos em cada minuto.	Bytes
NetworkMaxPacketsIn	A intermitência máxima por segundo de pacotes recebidos em cada minuto.	Contagem
NetworkMaxPacketsOut	A intermitência máxima por segundo de pacotes transmitidos em cada minuto.	Contagem

Métrica	Descrição	Unidade
SwapUsage	A quantidade de troca usada no host.	Bytes

Métricas para o Valkey e Redis OSS

O namespace Amazon ElastiCache inclui as seguintes métricas do Valkey e Redis OSS. Essas métricas são as mesmas quando se usa o mecanismo Valkey.

Com exceção de ReplicationLag, EngineCPUUtilization, SuccessfulWriteRequestLatency e SuccessfulReadRequestLatency, essas métricas são derivadas do comando info. Cada métrica é calculada no nível de nó de cache.

Para obter a documentação completa do info comando, consulte <http://valkey.io/commands/info>.

Consulte também

- [Métricas em nível de host](#)

Métrica	Description	Unidade
ActiveDefragHits	O número de realocações de valor por minuto executada pelo processo de desfragmenção ativo. Deriva da estatística active_defrag_hits no comando INFO .	Número
AuthenticationFailures	O número total de tentativas falhadas de autenticação para o Valkey ou Redis OSS usando o comando AUTH. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
BytesUsedForCache	O número total de bytes alocados pelo Valkey ou Redis OSS para todas as finalidades, inclusive o conjunto de dados, buffers, etc.	Bytes

Métrica	Description	Unidade
	Dimension: Tier=Memory para clusters do Valkey ou Redis OSS usando Classificação de dados em níveis no ElastiCache : o número total de bytes usados para cache por memória. Esse é o valor da estatística used_memory em INFO .	Bytes
	Dimension: Tier=SSD para clusters do Valkey ou Redis OSS usando Classificação de dados em níveis no ElastiCache : o número total de bytes usados para cache por SSD.	Bytes
BytesReadFromDisk	O número total de bytes lidos no disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis no ElastiCache .	Bytes
BytesWrittenToDisk	O número total de bytes gravados no disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis no ElastiCache .	Bytes
CacheHits	O número de buscas de chaves somente leitura bem-sucedidas no dicionário principal. Deriva da estatística keyspace_hits no comando INFO .	Contagem
CacheMisses	O número de buscas de chaves somente leitura malsucedidas no dicionário principal. Deriva da estatística keyspace_misses no comando INFO .	Contagem

Métrica	Description	Unidade
<code>CommandAuthorizationFailures</code>	O número total de tentativas falhadas por usuários para executar comandos que eles não têm permissão para chamar. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
<code>CacheHitRate</code>	Indica a eficiência de uso da instância Valkey ou Redis OSS. Se a proporção de cache for inferior a aproximadamente 0,8, isso significa que uma quantidade significativa de chaves são despejadas, expiradas ou não existem. Isso é calculado usando as estatísticas <code>cache_hits</code> e <code>cache_misses</code> da seguinte maneira: $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$.	Percentual
<code>ChannelAuthorizationFailures</code>	O número total de tentativas falhadas por usuários de acessar canais que eles não têm permissão para acessar. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme nesta métrica para detectar tentativas de acesso não autorizadas.	Contagem
<code>CurrConnections</code>	O número de conexões de clientes, excluindo conexões de réplicas de leitura. ElastiCache usa de 4 a 6 conexões para monitorar o cluster em cada caso. Deriva da estatística <code>connected_clients</code> no comando INFO .	Contagem

Métrica	Description	Unidade
CurrItems	O número de itens no cache. Deriva da estatística keyspace, somando todas as chaves em todo o keyspace.	Contagem
	Dimension: Tier=Memory para clusters usando Classificação de dados em níveis no ElastiCache . O número de itens em memória.	Contagem
	Dimension: Tier=SSD (unidades de estado sólido) para clusters usando Classificação de dados em níveis no ElastiCache . O número de itens em SSD.	Contagem
CurrVolatileItems	Número total de chaves em todos os bancos de dados que têm um ttl definido. Deriva da estatística expires, somando todas as chaves com um ttl definido em todo o keyspace.	Contagem
DatabaseCapacityUsagePercentage	Porcentagem da capacidade total de dados para o cluster que está em uso. Em instâncias com camadas de dados, a métrica é calculada como $(\text{used\_memory} - \text{mem\_not\_counted\_for\_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})$, onde <code>used_memory</code> e <code>maxmemory</code> são obtidas de INFO. Em todos os outros casos, a métrica é calculada usando <code>used_memory/maxmemory</code>.	Percentual

Métrica	Description	Unidade
DatabaseCapacityUsageCountedForEvictPercentage	Porcentagem da capacidade total de dados para o cluster que está em uso, excluindo a memória usada para sobrecarga e COB. Essa métrica é calculada como: $\frac{\text{used\_memory} - \text{mem\_not\_counted\_for\_evict}}{\text{maxmemory}}$ Em instâncias de dados em camadas, a métrica é calculada como: $\frac{(\text{used\_memory} + \text{SSD used})}{(\text{maxmemory} + \text{SSD total capacity})}$ onde <code>used_memory</code> e <code>maxmemory</code> são obtidos de INFO	Percentual
DatabaseMemoryUsagePercentage	Porcentagem de memória para o cluster que está em uso. É calculado usando <code>used_memory/maxmemory</code> de INFO.	Percentual
DatabaseMemoryUsageCountedForEvictPercentage	Porcentagem de memória para o cluster que está em uso, excluindo a memória usada para sobrecarga e COB. É calculado usando <code>used_memory-mem_not_counted_for_evict/maxmemory</code> de INFO.	Percentual

Métrica	Description	Unidade
DB0AverageTTL	Expõe o <code>avg_ttl</code> de DBO a partir da estatística <code>keyspace</code> do comando INFO . As réplicas não expiram as chaves; em vez disso, esperam que os nós primários expirem as chaves. Quando um nó primário expira uma chave (ou a remove por causa do LRU), ele sintetiza um DEL, que é transmitido para todas as réplicas. Assim sendo, DB0AverageTTL é 0 para nós de réplica, devido ao fato de que eles não expiram chaves e, portanto, não rastreiam TTL.	Milissegundos

Métrica	Description	Unidade
EngineCPUUtilization	Fornece utilização da CPU do thread do mecanismo Valkey ou Redis OSS. Como o Valkey e o Redis OSS são de thread único, você pode usar essa métrica para analisar a carga do próprio processo. A métrica EngineCPUUtilization fornece uma visibilidade mais precisa do processo. Você pode usá-la em conjunto com a métrica CPUUtilization . CPUUtilization expõe a utilização de CPU da instância do servidor como um todo, incluindo outros processos de sistema operacional e de gerenciamento. Para tipos de nós maiores com quatro v CPUs ou mais, use a EngineCPU Utilization métrica para monitorar e definir limites para escalabilidade.	Percentual

 Note

Em um ElastiCache host, os processos em segundo plano monitoram o host para fornecer uma experiência de banco de dados gerenciado. Esses processos em segundo plano podem ocupar uma parte significativa da workload da CPU. Isso não é significativo em hosts maiores com mais de dois CPUs v. Mas isso pode afetar hosts menores com 2v CPUs ou menos. Se você monitorar apenas a métrica EngineCPUUtilization , desconhecerá situações em que o host está sobrecarregado com alto uso da CPU do Valkey ou Redis OSS e alto uso da CPU dos processos de

Métrica	Description	Unidade
	monitoramento em segundo plano. Portanto, recomendamos monitorar a <code>CPUUtilization</code> métrica para hosts com dois v CPUs ou menos.	
Evictions	O número de chaves que foram removidas devido ao limite <code>maxmemory</code> . Deriva da estatística <code>evicted_keys</code> no comando INFO .	Contagem
GlobalDatastoreReplicationLag	Esse é o atraso entre o nó primário da região secundária e o nó primário da região primária. Para o Valkey ou Redis OSS habilitado para modo cluster, o atraso indica o atraso máximo entre os fragmentos.	Segundos
IamAuthenticationExpirations	O número total de conexões do Valkey ou Redis OSS expiradas autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
IamAuthenticationThrottling	O número total de solicitações AUTH ou HELLO do Valkey ou Redis OSS com controle de utilização autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
IsMaster	Indica se o nó é o nó primário do fragmento /cluster atual. A métrica pode ser 0 (não primária) ou 1 (primária).	Contagem

Métrica	Description	Unidade
KeyAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar chaves que eles não têm permissão para acessar. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
KeysTracked	O número de chaves que estão sendo monitoradas pelo monitoramento de chaves do Valkey ou Redis OSS como um percentual de <code>tracking-table-max-keys</code> . O monitoramento de chaves é usado para ajudar o cache do lado do cliente e notifica os clientes quando as chaves são modificadas.	Contagem
MemoryFragmentationRatio	Indica a eficiência na alocação de memória do mecanismo Valkey ou Redis OSS. Certos limites significarão comportamentos diferentes. O valor recomendado é ter fragmentação acima de 1,0. É calculada com base em <code>mem_fragmentation_ratio</code> statistic do comando INFO .	Número

Métrica	Description	Unidade
NewConnections	O número total de conexões que foram aceitas pelo servidor durante esse período. Deriva da estatística <code>total_connections_received</code> no comando INFO.  Note Se você estiver usando o ElastiCache Redis OSS versão 5 ou inferior, entre duas e quatro das conexões relatadas por essa métrica serão usadas ElastiCache para monitorar o cluster. No entanto, ao usar o ElastiCache Redis OSS versão 6 ou superior, as conexões usadas ElastiCache para monitorar o cluster não são incluídas nessa métrica.	Contagem
NumItemsReadFromDisk	O número total de itens recuperados do disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis no ElastiCache .	Contagem
NumItemsWrittenToDisk	O número total de itens gravados no disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis no ElastiCache .	Contagem
MasterLinkHealthStatus	Esse status tem dois valores: 0 ou 1. O valor 0 indica que os dados no nó ElastiCache primário não estão sincronizados com o Valkey ou o Redis OSS ativado. EC2 O valor de 1 indica que os dados não estão sincronizados. Para concluir a migração, use a operação CompleteMigration da API.	Booleano

Métrica	Description	Unidade
Reclaimed	O número total de eventos de expiração de chaves. Deriva da estatística <code>expired_keys</code> no comando INFO .	Contagem
ReplicationBytes	Para nós em uma configuração replicada, <code>ReplicationBytes</code> informa o número de bytes que a primária está enviando para todas as suas réplicas. Essa métrica é representante da carga de gravação no grupo de replicação. Deriva da estatística <code>master_repl_offset</code> no comando INFO .	Bytes
ReplicationLag	Essa métrica é aplicável somente para um nó de em execução como uma réplica de leitura. Ela representa o tempo decorrido, em segundos, até a réplica aplicar alterações do nó primário. Para o Valkey 7.2 e posteriores, e Redis OSS 5.0.6 e posteriores, o atraso pode ser medido em milissegundos.	Segundos
SaveInProgress	Essa métrica binária retorna 1 sempre que um salvamento de plano de fundo (com ou sem bifurcação) está em andamento e 0 caso não esteja. Um processo de salvamento de plano de fundo geralmente é usado durante snapshots e sincronizações. Essas operações podem prejudicar o desempenho. Usando a métrica <code>SaveInProgress</code> , você pode diagnosticar se uma performance prejudicada foi causada por um processo de salvamento de plano de fundo. Deriva da estatística <code>rdb_bgsave_in_progress</code> no comando INFO .	Booleano

Métrica	Description	Unidade
TrafficManagementActive	Indica se, ElastiCache para Redis, o OSS está gerenciando ativamente o tráfego ajustando o tráfego alocado para comandos de entrada, monitoramento ou replicação. O tráfego é gerenciado quando mais comandos são enviados ao nó do que podem ser processados pelo Valkey ou Redis OSS e é usado para manter a estabilidade e a operação ideal do mecanismo. Quaisquer pontos de dados de 1 podem indicar que o nó está subdimensionado para a workload que está sendo fornecida.  Note Se essa métrica permanecer ativa, avalie o cluster para decidir se é necessário aumentar a escala verticalmente ou horizontalmente. As métricas relacionadas incluem NetworkBandwidthOutAllowanceExceeded e EngineCPU Utilization .	Booleano
SuccessfulWriteRequestLatency	Latência de solicitações de gravação bem-sucedidas. Estatísticas válidas: Average (Média), Sum (Soma), Min (Mínimo), Max (Máximo), Sample Count (Contagem de amostras), qualquer percentil entre p0 e p100. A contagem de amostras inclui somente os comandos que foram executados com sucesso.	Microsegundos

Métrica	Description	Unidade
SuccessfulReadRequestLatency	Latência de solicitações de leitura bem-sucedidas. Estatísticas válidas: Average (Média), Sum (Soma), Min (Mínimo), Max (Máximo), Sample Count (Contagem de amostras), qualquer percentil entre p0 e p100. A contagem de amostras inclui somente os comandos que foram executados com sucesso.	Microssegundos
ErrorCount	O número total de comandos com falha durante o período especificado. Estatísticas válidas: média, soma, máx. e mín.	Contagem
SearchNumberOfIndexes	Número de índices criados	Contagem
SearchTotalIndexedDocuments	Número total de chaves em todos os índices	Contagem
SearchUsedMemoryBytes	Número de bytes de memória consumidos em todas as estruturas de dados de pesquisa	Bytes

CPUUtilization Disponibilidade do motor

AWS As regiões listadas a seguir estão disponíveis em todos os tipos de nós compatíveis.

Região	Nome da região
us-east-2	Leste dos EUA (Ohio)
us-east-1	Leste dos EUA (Norte da Virgínia)
us-west-1	Oeste dos EUA (N. da Califórnia)
us-west-2	Oeste dos EUA (Oregon)

Região	Nome da região
ap-northeast-1	Ásia-Pacífico (Tóquio)
ap-northeast-2	Ásia-Pacífico (Seul)
ap-northeast-3	Asia Pacific (Osaka)
ap-east-1	Ásia-Pacífico (Hong Kong)
ap-south-1	Asia Pacific (Mumbai)
ap-southeast-1	Ásia-Pacífico (Singapura)
ap-southeast-2	Ásia-Pacífico (Sydney)
ap-southeast-3	Ásia-Pacífico (Jacarta)
ca-central-1	Canadá (Central)
cn-north-1	China (Pequim)
cn-northwest-2	China (Ningxia)
me-south-1	Oriente Médio (Bahrein)
eu-central-1	Europa (Frankfurt)
eu-west-1	Europa (Irlanda)
eu-west-2	Europa (Londres)
eu-west-3	UE (Paris)
eu-south-1	Europa (Milão)
af-south-1	África (Cidade do Cabo)
eu-north-1	Europa (Estocolmo)
sa-east-1	América do Sul (São Paulo)

Região	Nome da região
us-gov-west-1	AWS GovCloud (Oeste dos EUA)
us-gov-east-1	AWS GovCloud (Leste dos EUA)

A seguir estão agregações de determinados tipos de comandos, derivados de `info commandstats`: A seção `commandstats` fornece estatísticas baseadas no tipo de comando, incluindo o número de chamadas, o tempo total de CPU consumido por esses comandos e a CPU média consumida por execução de comando. Para cada tipo de comando, é adicionada a seguinte linha: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

As métricas de latência listadas a seguir são calculadas com a estatística `commandstats` de [INFO](#). Elas são calculadas da seguinte maneira: $\text{delta}(\text{usec})/\text{delta}(\text{calls})$. O `delta` é calculado como a diferença dentro de um minuto. A latência é definida como o tempo gasto pela CPU ElastiCache para processar o comando. Observe que, para clusters que usam classificação de dados por nível, o tempo necessário para buscar itens no SSD não está incluído nessas medições.

Para obter uma lista completa dos comandos disponíveis, consulte [comandos](#) na documentação do Valkey.

Métrica	Description	Unidade
<code>ClusterBasedCmds</code>	O número total de comandos que são baseados em cluster. Deriva da estatística <code>commandstats</code> , somando todos os comandos que atuam em um cluster (<code>cluster slot</code> , <code>cluster info</code> etc.).	Contagem
<code>ClusterBasedCmdsLatency</code>	Latência de comandos baseados em cluster.	Microsegundos
<code>EvalBasedCmds</code>	O número total de comandos para comandos baseados em avaliação. É derivado da estatística <code>commandstats</code> , pela soma de <code>eval</code> , <code>evalsha</code> .	Contagem

Métrica	Description	Unidade
EvalBasedCmdsLatency	Latência de comandos baseados em avaliação.	Microssegundos
GeoSpatialBasedCmds	O número total de comandos para comandos baseados em dados geoespaciais. É derivado da estatística <code>commandstats</code> . Ele é derivado somando todos o tipos de comandos geo: <code>geoadd</code> , <code>geodist</code> , <code>geohash</code> , <code>geopos</code> , <code>georadius</code> , e <code>georadiusbymember</code> .	Contagem
GeoSpatialBasedCmdsLatency	Latência de comandos baseados em dados geoespaciais.	Microssegundos
GetTypeCmds	O número total de comandos do tipo read-only . É derivado da estatística <code>commandstats</code> , somando todos os comandos do tipo read-only (<code>get</code> , <code>hget</code> , <code>scard</code> , <code>lrange</code> , etc.)	Contagem
GetTypeCmdsLatency	Latência de comandos de leitura.	Microssegundos
HashBasedCmds	O número total de comandos baseados em hash. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em um ou mais hashes (<code>hget</code> , <code>hkeys</code> , <code>hvals</code> , <code>hdel</code> , etc.).	Contagem
HashBasedCmdsLatency	Latência de comandos baseados em hash.	Microssegundos
HyperLogLogBasedCmds	O número total de comandos baseados em HyperLogLog . É derivado da estatística <code>commandstats</code> , somando todos os comandos do tipo pf (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> , etc.).	Contagem

Métrica	Description	Unidade
HyperLogLogBasedCmdsLatency	Latência dos comandos HyperLogLog baseados.	Microssegundos
JsonBasedCmds	O número total de comandos JSON, incluindo comandos de leitura e gravação. É derivado da estatística <code>commandstats</code> , somando todos os comandos JSON que atuam em chaves JSON.	Contagem
JsonBasedCmdsLatency	Latência de todos os comandos JSON, incluindo comandos de leitura e gravação.	Microssegundos
JsonBasedGetCmds	O número total de comandos JSON somente leitura. É derivado da estatística <code>commandstats</code> , somando todos os comandos de leitura JSON que atuam em chaves JSON.	Contagem
JsonBasedGetCmdsLatency	Latência dos comandos JSON somente leitura.	Microssegundos
JsonBasedSetCmds	O número total de comandos de gravação JSON. É derivado da estatística <code>commandstats</code> , somando todos os comandos de gravação JSON que atuam em chaves JSON.	Contagem
JsonBasedSetCmdsLatency	Latência dos comandos de gravação JSON.	Microssegundos
KeyBasedCmds	O número total de comandos baseados em chave. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em uma ou mais chaves em várias estruturas de dados (<code>del</code> , <code>expire</code> , <code>rename</code> , etc.).	Contagem
KeyBasedCmdsLatency	Latência de comandos baseados em chave.	Microssegundos

Métrica	Description	Unidade
ListBasedCmds	O número total de comandos baseados em lista. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em uma ou mais listas (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> , etc.).	Contagem
ListBasedCmdsLatency	Latência de comandos baseados em lista.	Microssegundos
NonKeyTypeCmds	O número total de comandos não baseados em chave. É derivado da estatística <code>commandstats</code> , somando todos os comandos que não atuam em uma chave; por exemplo, <code>acl</code> , <code>dbsize</code> ou <code>info</code> .	Contagem
NonKeyTypeCmdsLatency	Latência dos non-key-based comandos.	Microssegundos
PubSubBasedCmds	O número total de comandos para pub/sub funcionalidade. Isso é derivado das <code>commandstats</code> estatísticas pela soma de todos os comandos usados para pub/sub funcionalidade: <code>psubscribe</code> , <code>publishpubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>sunsubscribe</code> , <code>spublish</code> , <code>subscribe</code> , e <code>unsubscribe</code> .	Contagem
PubSubBasedCmdsLatency	Latência de comandos baseados em pub/sub.	Microssegundos
SetBasedCmds	O número total de comandos que são baseados em conjuntos. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em um ou mais conjuntos (<code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>union</code> , etc.).	Contagem

Métrica	Description	Unidade
SetBasedCmdsLatency	Latência de comandos baseados em conjunto.	Microssegundos
SetTypeCmds	O número total de tipos de comando write. É derivado da estatística <code>commandstats</code> , somando todos os tipos de comando mutative que operam em dados (set, hset, sadd, lpop, etc.)	Contagem
SetTypeCmdsLatency	Latência de comandos de gravação.	Microssegundos
SortedSetBasedCmds	O número total de comandos que são classificados com base em conjuntos. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em um ou mais conjuntos classificados (zcount, zrange, zrank, zadd, etc.).	Contagem
SortedSetBasedCmdsLatency	Latência de comandos baseados em classificação.	Microssegundos
StringBasedCmds	O número total de comandos baseados em string. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em uma ou mais strings (strlen, setex, setrange, etc.).	Contagem
StringBasedCmdsLatency	Latência de comandos baseados em string.	Microssegundos
StreamBasedCmds	O número total de comandos que são baseados em fluxo. É derivado da estatística <code>commandstats</code> , somando todos os comandos que atuam em um ou mais tipos de dados de fluxos (xrange, xlen, xadd, xdel, etc.).	Contagem

Métrica	Description	Unidade
StreamBasedCmdsLatency	Latência de comandos baseados em fluxo.	Microssegundos
SearchBasedCmds	O número total de comandos Search, incluindo comandos de leitura e gravação. Isso é derivado da estatística commandstats pela soma de todos os comandos Search.	Contagem
SearchBasedCmdsLatency	Latência de todos os comandos Search, incluindo comandos de leitura e gravação.	Microssegundos
SearchBasedGetCmds	O número total de comandos Search somente leitura. Isso é derivado da estatística commandstats pela soma de todos os comandos Search de leitura.	Contagem
SearchBasedGetCmdsLatency	Latência dos comandos Search somente leitura.	Microssegundos
SearchBasedSetCmds	O número total de comandos de gravação Search. Isso é derivado da estatística commandstats pela soma de todos os comandos Search de gravação.	Contagem
SearchBasedSetCmdsLatency	Latência dos comandos Search de leitura.	Microssegundos

Métricas para o Memcached

O namespace AWS/ElastiCache inclui as métricas do Memcached a seguir.

O namespace da AWS/ElastiCache inclui as métricas a seguir derivadas do comando stats do Memcached. Cada métrica é calculada no nível de nó de cache.

Consulte também

- [Métricas em nível de host](#)

Métrica	Descrição	Unidade
BytesReadIntoMemcached	O número de bytes que foram lidos da rede pelo nó de cache.	Bytes
BytesUsedForCacheItems	O número de bytes usados para armazenar itens de cache.	Bytes
BytesWrittenOutFromMemcached	O número de bytes que foram gravados na rede pelo nó de cache.	Bytes
CasBadval	O número de solicitações de CAS (verificar e definir) recebidas pelo cache em que o valor de Cas não correspondeu ao valor de Cas armazenado.	Contagem
CasHits	O número de solicitações de Cas recebidas pelo cache em que a chave solicitada foi encontrada e o valor de Cas correspondia.	Contagem
CasMisses	O número de solicitações Cas recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
CmdFlush	O número de comandos flush recebidos pelo cache.	Contagem
CmdGet	O número de comandos get recebidos pelo cache.	Contagem
CmdSet	O número de comandos set recebidos pelo cache.	Contagem
CurrConnections	A contagem do número de conexões do cache em determinado momento. O ElastiCache usa de duas a três conexões para monitorar o cluster.	Contagem

Métrica	Descrição	Unidade
	Além do mencionado acima, o Memcached cria um número de conexões internas iguais ao dobro dos threads usados para o tipo de nó. A contagem de threads para os vários tipos de nó pode ser vista no <code>Nodetype Specific Parameters</code> do grupo de parâmetros aplicável. O total de conexões é a soma das conexões de cliente, as conexões para monitoramento e as conexões internas mencionadas acima.	
<code>CurrItems</code>	A contagem do número de itens atualmente armazenados no cache.	Contagem
<code>DecrHits</code>	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
<code>DecrMisses</code>	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
<code>DeleteHits</code>	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
<code>DeleteMisses</code>	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
<code>Evictions</code>	O número de itens não expirados removidos pelo cache para permitir espaço para novas gravações.	Contagem

Métrica	Descrição	Unidade
GetHits	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
GetMisses	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
IncrHits	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
IncrMisses	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
Reclaimed	O número de itens expirados removidos pelo cache para permitir espaço para novas gravações.	Contagem

Para o Memcached 1.4.14, as seguintes métricas adicionais são fornecidas.

Métrica	Descrição	Unidade
BytesUsedForHash	O número de bytes atualmente usados por tabelas de hash.	Bytes
CmdConfigGet	O número cumulativo de solicitações config get.	Contagem
CmdConfigSet	O número cumulativo de solicitações config set.	Contagem
CmdTouch	O número cumulativo de solicitações touch.	Contagem
CurrConfig	O número atual de configurações armazenadas.	Contagem

Métrica	Descrição	Unidade
EvictedUnfetched	O número de itens válidos removidos do cache menos usado recentemente (LRU) que nunca foram tocados após serem definidos.	Contagem
ExpiredUnfetched	O número de itens expirados recuperados do LRU que nunca foram tocados após serem definidos.	Contagem
SlabsMoved	O número total de páginas de prancha que foram movidas.	Contagem
TouchHits	O número de chaves que foram tocadas e receberam um novo horário de expiração.	Contagem
TouchMisses	O número de itens que foram tocados, mas não foram encontrados.	Contagem

O namespace da AWS/ElastiCache inclui as métricas a seguir calculadas no nível de cache.

Métrica	Descrição	Unidade
NewConnections	O número de novas conexões recebidas pelo cache. Isso é derivado da estatística <code>total_connections</code> do Memcached registrando a alteração em <code>total_connections</code> durante um período. Sempre será pelo menos 1, devido a uma conexão reservada para um ElastiCache.	Contagem
NewItems	O número de novos itens armazenados pelo cache. Isso é derivado da estatística <code>total_items</code> do Memcached registrando a alteração em <code>total_items</code> ao longo de um período.	Contagem
UnusedMemory	A quantidade de memória não usada pelos dados. Isso é derivado das estatísticas	Bytes

Métrica	Descrição	Unidade
	limit_maxbytes e bytes do Memcached subtraindo bytes de limit_maxbytes. Como a sobrecarga do Memcached usa memória além daquela usada pelos dados, UnusedMemory não deve ser considerada como a quantidade de memória disponível para dados adicionais. Você pode passar por remoções, mesmo que você ainda tenha alguma memória não utilizada. Para obter informações mais detalhadas, consulte Memcached item memory usage.	

Que métricas devo monitorar?

As CloudWatch métricas a seguir oferecem uma boa visão ElastiCache do desempenho. Na maioria dos casos, recomendamos que você defina CloudWatch alarmes para essas métricas para que você possa tomar medidas corretivas antes que ocorram problemas de desempenho.

Métricas para monitorar

- [CPUUtilization](#)
- [Motor CPUUtilization](#)
- [SwapUsage \(Valkey e Redis OSS\)](#)
- [Evictions](#)
- [CurrConnections](#)
- [Memória \(Valkey e Redis OSS\)](#)
- [Rede](#)
- [Latência](#)
- [Replicação](#)
- [Gerenciamento de tráfego \(Valkey e Redis OSS\)](#)

CPUUtilization

Essa é uma métrica em nível de host relatada como uma porcentagem. Para obter mais informações, consulte [Métricas em nível de host](#).

Valkey e Redis OSS

Para tipos de nós menores com 2v CPUs ou menos, use a CPUUtilization métrica para monitorar sua carga de trabalho.

De modo geral, sugerimos que você defina o limite para 90% da CPU disponível. Como o Valkey e o Redis OSS têm thread único, o valor efetivo do limite deve ser calculado como uma fração da capacidade total do nó. Por exemplo, suponha que você esteja usando um tipo de nó com dois núcleos. Nesse caso, o limite para CPUUtilization seria 90/2 ou 45%.

Você precisará determinar seu próprio limite, com base no número de núcleos no nó de cache que você está usando. Se você exceder esse limite e sua workload principal for de solicitações de leitura, escale seu cluster adicionando réplicas de leitura. Se a workload principal é de solicitações de gravação, dependendo da configuração do cluster, recomendamos que você:

- Clusters do Valkey ou Redis OSS (modo cluster desabilitado): aumente a escala verticalmente usando um tipo de instância com cache maior.
- Clusters do Valkey ou Redis OSS (modo cluster habilitado): adicione mais fragmentos para distribuir a workload de gravação entre nós primários adicionais.

Tip

Em vez de usar a métrica `CPUUtilization` de nível de host, os usuários do Valkey e Redis OSS podem usar a métrica `EngineCPUUtilization`, que informa a porcentagem de uso no núcleo do mecanismo Valkey ou Redis OSS. Para ver se essa métrica está disponível nos seus nós e obter mais informações, consulte [Métricas do Valkey e Redis OSS](#).

Para tipos de nós maiores com 4v CPUs ou mais, talvez você queira usar a `EngineCPUUtilization` métrica, que relata a porcentagem de uso no núcleo do mecanismo Valkey ou Redis OSS. Para ver se essa métrica está disponível nos seus nós e obter mais informações, consulte [Métricas do Redis OSS](#).

Memcached

Como o Memcached é multi-threaded, essa métrica pode atingir até 90%. Se você exceder esse limite, aumente a escala de seu cluster usando um tipo de nó de cache maior ou aumente a escala horizontalmente adicionando mais nós de cache.

Motor CPUUtilization

Para tipos de nós maiores com 4v CPUs ou mais, talvez você queira usar a `EngineCPUUtilization` métrica, que relata a porcentagem de uso no núcleo do mecanismo Redis OSS. Para ver se essa métrica está disponível nos seus nós e obter mais informações, consulte [Métricas do Valkey e Redis OSS](#).

Para obter mais informações, consulte a CPU seção [Monitoramento das melhores práticas com o Amazon ElastiCache for Redis OSS usando a Amazon. CloudWatch](#)

SwapUsage (Valkey e Redis OSS)

Esta é uma métrica em nível de host relatada em bytes. Para obter mais informações, consulte [Métricas em nível de host](#).

A FreeableMemory CloudWatch métrica próxima de 0 (ou seja, abaixo de 100 MB) ou a SwapUsage métrica maior que a FreeableMemory métrica indica que um nó está sob pressão de memória. Se isso ocorrer, consulte os seguintes tópicos:

- [Garantir que você tem memória suficiente para criar um snapshot do Valkey ou Redis OSS](#)
- [Gerenciamento de memória reservada para Valkey e Redis OSS](#)

Evictions

Esta é uma métrica do mecanismo de cache. Recomendamos que você determine seu próprio limite de alarme para essa métrica com base nas necessidades do seu aplicativo.

Se você estiver usando o Memcached e exceder o limite escolhido, aumente a escala do seu cluster usando um tipo de nó maior ou aumente a escala horizontalmente adicionando mais nós.

CurrConnections

Esta é uma métrica do mecanismo de cache. Recomendamos que você determine seu próprio limite de alarme para essa métrica com base nas necessidades do seu aplicativo.

Um número crescente de CurrConnections pode indicar um problema com seu aplicativo; você precisará investigar o comportamento do aplicativo para resolver esse problema.

Para obter mais informações, consulte a seção Conexões em [Monitoramento das melhores práticas com o Amazon ElastiCache for Redis OSS usando a Amazon. CloudWatch](#)

Memória (Valkey e Redis OSS)

A memória é um aspecto central do Valkey e do Redis OSS. Compreender a utilização da memória do seu cluster é necessário para evitar a perda de dados e acomodar o crescimento futuro do seu conjunto de dados. Estatísticas sobre a utilização de memória de um nó estão disponíveis na seção de memória do comando [INFO](#).

Para obter mais informações, consulte a seção Memória em [Monitoramento das melhores práticas com o Amazon ElastiCache for Redis OSS usando a Amazon. CloudWatch](#)

Rede

Um dos fatores determinantes para a capacidade de largura de banda da rede do cluster é o tipo de nó selecionado. Para obter mais informações sobre a capacidade de rede do seu nó, consulte os [ElastiCache preços da Amazon](#).

Para obter mais informações, consulte a seção Rede em [Monitoramento das melhores práticas com o Amazon ElastiCache for Redis OSS usando a Amazon CloudWatch](#)

Latência

A medição do tempo de resposta ElastiCache para uma instância do Valkey pode ser abordada de várias maneiras, dependendo do nível de granularidade necessário. Os principais estágios que contribuem para o tempo geral de resposta do lado do servidor ElastiCache para o Valkey são o pré-processamento do comando, a execução do comando e o pós-processamento do comando.

As métricas de latência específicas do comando derivadas do comando Valkey [INFO](#), como `GetTypeCmdsLatency` e `SetTypeCmdsLatency` métrica, se concentram especificamente na execução da lógica de comando principal do comando Valkey. Essas métricas serão úteis se seu caso de uso for definir o tempo de execução do comando ou as latências agregadas por estrutura de dados.

A latência mede `SuccessfulWriteRequestLatency` e `SuccessfulReadRequestLatency` mede o tempo total que o motor ElastiCache for Valkey leva para responder a uma solicitação.

Note

Valores inflados para as métricas `SuccessfulWriteRequestLatency` e `SuccessfulReadRequestLatency` podem ocorrer ao usar o pipeline do Valkey com `CLIENT REPLY` habilitado no cliente do Valkey. A criação de pipeline do Valkey é uma técnica para melhorar o desempenho emitindo vários comandos ao mesmo tempo, sem esperar pela resposta de cada comando individual. Para evitar valores inflados, recomendamos configurar seu cliente do Valkey para gerar comandos do pipeline com [CLIENT REPLY OFF](#).

Para obter mais informações, consulte a seção Latência em [Monitoramento das melhores práticas com a Amazon ElastiCache usando a Amazon CloudWatch](#).

Replicação

O volume de dados que está sendo replicado é visível através da métrica `ReplicationBytes`. Embora essa métrica seja representante da carga de gravação no grupo de replicação, ela não fornece insights sobre a integridade da replicação. Para este efeito, você pode usar a métrica `ReplicationLag`.

Para obter mais informações, consulte a seção Replicação em [Monitoramento das melhores práticas com o Amazon ElastiCache for Redis OSS usando](#) a Amazon. CloudWatch

Gerenciamento de tráfego (Valkey e Redis OSS)

ElastiCache for Redis OSS gerencia automaticamente o tráfego em relação a um nó quando mais comandos de entrada são enviados para o nó do que os que podem ser processados pelo Valkey ou pelo Redis OSS. Isso é feito para manter a operação e estabilidade ideais do mecanismo.

Quando o tráfego for gerenciado ativamente em um nó, a métrica `TrafficManagementActive` emitirá pontos de dados de 1. Isso indica que o nó está subdimensionado para a workload que está sendo fornecida. Se essa métrica permanecer em 1 por longos períodos, avalie o cluster para decidir se é necessário aumentar a escala verticalmente ou horizontalmente.

Para obter mais informações, consulte a métrica `TrafficManagementActive` na página [Métricas](#).

Escolher estatísticas e períodos de métricas

Embora o CloudWatch permita que você escolha qualquer estatística e período para cada métrica, nem todas as combinações são úteis. Por exemplo, as estatísticas Average, Minimum e Maximum para CPUUtilization são úteis, mas a estatística Sum não é.

Todas as amostras do ElastiCache são publicadas por uma duração de 60 segundos para cada nó de cache individual. Para qualquer período de 60 segundos, uma métrica de nó de cache apenas conterá uma única amostra.

Para obter mais informações sobre como recuperar métricas para seus nós de cache, consulte [Monitorando métricas de CloudWatch clusters e nós](#).

Monitorando métricas de CloudWatch clusters e nós

ElastiCache e CloudWatch são integrados para que você possa reunir uma variedade de métricas. Você pode monitorar essas métricas usando CloudWatch o.

Note

Os exemplos a seguir exigem as ferramentas de linha de CloudWatch comando. Para obter mais informações CloudWatch e fazer o download das ferramentas para desenvolvedores, consulte a [página CloudWatch do produto](#).

Os procedimentos a seguir mostram como usar CloudWatch para coletar estatísticas de espaço de armazenamento de um cluster na última hora.

Note

Os valores StartTime e EndTime fornecidos nos exemplos abaixo são para fins ilustrativos. Você deve substituir valores apropriados de tempo de início e término para seus nós de cache.

Para obter informações sobre ElastiCache limites, consulte [Limites AWS de serviço](#) para ElastiCache.

Monitorando métricas de CloudWatch clusters e nós (console)

Para coletar estatísticas de utilização de CPU para um cluster de cache

1. Faça login no Console de gerenciamento da AWS e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Selecione os nós de cache para os quais você deseja visualizar métricas.

 Note

Selecionar mais de 20 nós desabilita as métricas de visualização no console.

- Na página Clusters de Cache do AWS Management Console, clique no nome de um ou mais clusters.
- A página de detalhes do cluster é exibida.
- Clique na guia Nodes na parte superior da janela.
 - Na guia Nodes da janela de detalhes, selecione os nós de cache para os quais você deseja visualizar as métricas.

A página de detalhes do cluster é exibida.

- d. Clique na métrica CPU Utilization.

O CloudWatch console será aberto, exibindo as métricas selecionadas. Você pode usar as caixas de listagem suspensas **Statistic** e **Period** e a guia **Time Range** para alterar as métricas exibidas.

Monitorando métricas de CloudWatch clusters e nós usando a CloudWatch CLI

Para coletar estatísticas de utilização de CPU para um cluster de cache

- Para 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
```

Para 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
```

Monitorando métricas de CloudWatch clusters e nós usando a CloudWatch API

Para coletar estatísticas de utilização de CPU para um cluster de cache

- Chame a CloudWatch API `GetMetricStatistics` com os seguintes parâmetros (observe que os horários de início e término são mostrados apenas como exemplos; você precisará substituir os horários de início e término apropriados):
 - `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=mycacheclass,CacheNodeId=0002`

Example

```
http://monitoring.amazonaws.com/  
  ?Action=GetMetricStatistics  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &StartTime=2018-07-05T00:00:00  
  &EndTime=2018-07-06T23:59:00  
  &Period=3600  
  &Statistics.member.1=Average  
  &Dimensions.member.1="CacheClusterId=mycachecluster"  
  &Dimensions.member.2="CacheNodeId=0002"  
  &Namespace=&AWS;/ElastiCache  
  &MeasureName=CPUUtilization  
  &Timestamp=2018-07-07T17%3A48%3A21.746Z  
  &AWS;AccessKeyId=<&AWS; Access Key ID>  
  &Signature=<Signature>
```

Cotas do ElastiCache

Sua conta da AWS possui cotas padrão, anteriormente chamadas de limites, para cada produto da AWS. A menos que especificado de outra forma, cada cota é específica da região. Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Para visualizar as cotas do ElastiCache, abra o [console Service Quotas](#) (Cotas de serviço). No painel de navegação, selecione serviços da AWS e ElastiCache.

Para solicitar o aumento da cota, consulte [Requesting a Quota Increase](#) (Solicitação de aumento de cota) no Guia do usuário do Service Quotas. Se a cota ainda não estiver disponível no Service Quotas, use o [formulário de aumento de limite](#).

Sua conta da AWS tem as seguintes cotas relacionadas ao ElastiCache.

Name	Valor padrão	Descrição	Nome da métrica
Nós por região	300	O número máximo de nós entre todos os clusters em uma região. Essa cota se aplica aos nós não reservados em determinada região. Você pode ter até 300 nós não reservados na mesma região.	NodesPerRegion
Nós por cluster (Memcached)	60	O número máximo de nós em um cluster específico do Memcached.	Memcached NodesPerCluster
Nós por cluster (modo cluster habilitado)	90	O número máximo de nós em um cluster específico do Redis OSS ou Valkey.	NodesPerCluster

Name	Valor padrão	Descrição	Nome da métrica
Grupos de parâmetros por região	300	O número máximo de grupos de parâmetros que você pode criar em uma Região.	ParameterGroup
Grupos de sub-redes por região	300	O número máximo de grupos de sub-rede que você pode criar em uma Região.	SubnetGroup
Sub-redes por grupo de sub-rede	20	O número máximo de sub-redes que você pode definir para um grupo de sub-rede.	SubnetsPerSubnetGroup
Caches com tecnologia sem servidor por região	40	O número máximo de caches de tecnologia sem servidor em uma região.	ServerlessCache
Instantâneos sem servidor por dia por cache	24	O número máximo de instantâneos de tecnologia sem servidor que você pode tirar por dia para cada cache sem de tecnologia sem servidor.	SnapshotsPerDayPerCache
Usuários por região	2000	O número máximo total de usuários que você pode criar em uma Região.	Usuário

Name	Valor padrão	Descrição	Nome da métrica
Grupos de usuários por região	200	O número máximo de grupos de usuários que você pode criar em uma Região.	UserGroup
Usuários por grupo de usuários	100	O número máximo de usuários que você pode definir para um grupo de usuários.	UsersPerUserGroup

Referência

Os tópicos desta seção abrangem o trabalho com a API do Amazon ElastiCache e a seção do ElastiCache da AWS CLI. Também estão incluídas nesta seção mensagens de erro comuns e notificações de serviço.

- [Uso da API do ElastiCache](#)
- [Referência da API do ElastiCache](#)
- [Seção ElastiCache da Referência da AWS CLI](#)
- [Mensagens ElastiCache de erro da Amazon](#)
- [Notificações](#)

Uso da API do ElastiCache

Esta seção fornece descrições orientadas a tarefas sobre como usar e implementar operações do ElastiCache. Para obter uma descrição completa dessas operações, consulte a [Referência de API do Amazon ElastiCache](#).

Tópicos

- [Como usar a API de consulta](#)
- [Bibliotecas disponíveis](#)
- [Solução de problemas de aplicações](#)

Como usar a API de consulta

Parâmetros de consulta

As solicitações baseadas em consulta HTTP são solicitações HTTP que usam o verbo HTTP GET ou POST e um parâmetro de consulta chamado `Action`.

Cada solicitação de consulta deve incluir alguns parâmetros comuns para lidar com a autenticação e a seleção de uma ação.

Algumas operações levam listas de parâmetros. Essas listas são especificadas usando a notação `param.n`. Os valores de *n* são inteiros a partir de 1.

Autenticação de solicitação de consulta

Só é possível enviar solicitações de consulta por meio de HTTPS, e é preciso incluir uma assinatura em todas as solicitações de consulta. Esta seção descreve como criar a assinatura. O método descrito no procedimento a seguir é conhecido como versão de assinatura 4.

As etapas básicas a seguir são usadas para autenticar as solicitações à AWS. Isso presume que você esteja registrado na AWS e possua um ID de chave de acesso e uma chave de acesso secreta.

Processo de autenticação de consulta

1. O remetente elabora uma solicitação à AWS.
2. O remetente calcula a assinatura da solicitação, um hash codificado para o HMAC (Hash-based Message Authentication Code) com uma função de hash SHA-1, conforme definido na próxima seção deste tópico.
3. O remetente da solicitação envia os dados da solicitação, a assinatura e o ID da chave de acesso (o identificador da chave de acesso secreta usada) à AWS.
4. AWSA usa o ID de chave de acesso para pesquisar pela chave de acesso secreta.
5. AWSA gera uma assinatura a partir dos dados da solicitação e da chave de acesso secreta usando o mesmo algoritmo usado para calcular a assinatura enviada na solicitação.
6. Se as assinaturas coincidirem, a solicitação será considerada autêntica. Se a comparação falhar, a solicitação será descartada e a AWS retornará uma resposta de erro.

Note

Se uma solicitação contiver um parâmetro `Timestamp`, a assinatura calculada para a solicitação expirará 15 minutos após o valor.

Se uma solicitação contiver um parâmetro `Expires`, a assinatura expirará no horário especificado pelo parâmetro `Expires`.

Para calcular a assinatura da solicitação

1. Crie a query string canonizada de que você precisará posteriormente neste procedimento:

- a. Classifique os componentes query string UTF-8 por nome do parâmetro com o ordenamento natural de bytes. Os parâmetros podem vir do URI GET ou do corpo POST (quando Content-Type for application/x-www-form-urlencoded).
 - b. Codificar em URL o nome do parâmetro e os valores de acordo com as seguintes regras:
 - i. Não codificar em URL nenhum dos caracteres não reservados que definem o RFC 3986. Esses caracteres não reservados são A–Z, a–z, 0–9, hífen (-), sublinhado (_), ponto (.) e til (~).
 - ii. Codificar em percentual todos os outros caracteres com %XY, onde X e Y são caracteres hexadecimais de 0 a 9 e maiúsculas de A a F.
 - iii. Codificar em percentual os caracteres UTF-8 estendidos na forma %XY%ZA...
 - iv. Codificar em percentual o caractere de espaço como %20 (e não +, como em esquemas de codificação comuns).
 - c. Separe os nomes de parâmetro codificados a partir de seus valores codificados com o sinal de igual (=) (caractere ASCII 61), mesmo se o valor do parâmetro estiver vazio.
 - d. Separe os pares de nome-valor por um "&" (e comercial) (código 38 em ASCII).
2. Crie a string para assinar de acordo com a seguinte pseudogramática ("\\n" representa uma nova linha em ASCII).

```
StringToSign = HTTPVerb + "\\n" +  
ValueOfHostHeaderInLowercase + "\\n" +  
HTTPRequestURI + "\\n" +  
CanonicalizedQueryString <from the preceding step>
```

O componente HTTPRequestURI é o componente de caminho absoluto HTTP do URI até a query string (não incluída). Se o HTTPRequestURI estiver vazio, use uma barra (/).

3. Calcule um HMAC compatível com RFC 2104 com a string recém-criada, sua chave de acesso secreta como chave e SHA256 ou SHA1 como algoritmo de hash.

Para obter mais informações, consulte <https://www.ietf.org/rfc/rfc2104.txt>.

4. Converta o valor resultante para base64.
5. Inclua o valor como o valor do parâmetro Signature na solicitação.

Por exemplo, a seguir você encontra um exemplo de solicitação (as quebras de linha foram adicionadas para maior clareza).

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01
```

Quanto à string de consulta anterior, você calcularia a assinatura HMAC na seguinte string.

```
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:
```

O resultado é a seguinte solicitação assinada.

```
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
```

Para obter informações detalhadas sobre o processo de assinatura e o cálculo da assinatura da solicitação, consulte o tópico [Processo de assinatura do Signature Version 4](#) e seus subtópicos.

Bibliotecas disponíveis

AWSA fornece kits de desenvolvimento de software (SDKs) para desenvolvedores de software que preferem criar aplicações usando APIs específicas de linguagem em vez da API de consulta. Esses SDKs oferecem as funções básicas (não incluídas nas APIs), como autenticação de solicitação, novas tentativas de solicitação e processamento de erros, para que você possa começar a usar com mais facilidade. SDKs e recursos adicionais estão disponíveis para as seguintes linguagens de programação:

- [Java](#)
- [Windows and .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Para obter informações sobre outras linguagens, consulte [Código e bibliotecas de exemplo](#).

Solução de problemas de aplicações

O ElastiCache fornece erros específicos e descritivos para ajudar você a solucionar problemas enquanto interage com a API do ElastiCache.

Recuperação de erros

Normalmente, espera-se que o aplicativo verifique se uma solicitação gerou um erro antes que você precise processar os resultados. A maneira mais fácil de descobrir se ocorreu um erro é procurar por um nó de `Error` na resposta da API do ElastiCache.

A sintaxe XPath apresenta uma maneira simples de procurar pela presença de um nó `Error`, bem como uma maneira fácil de recuperar o código e a mensagem de erro. O snippet de código a seguir

usa Perl e o módulo XML::XPath para determinar se ocorreu um erro durante uma solicitação. Caso tenha ocorrido, o código imprimirá o primeiro código de erro e a mensagem na resposta.

```
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"; }
```

Dicas de solução de problemas

Recomendamos os seguintes processos para diagnosticar e resolver problemas com a API do ElastiCache.

- Verifique se o ElastiCache está funcionando corretamente.

Para fazer isso, basta abrir uma janela do navegador e enviar uma solicitação de consulta para o serviço do ElastiCache (como <https://elasticache.amazonaws.com>). Uma `MissingAuthenticationTokenException` ou um Erro do servidor interno 500 confirma que o serviço está disponível e respondendo a solicitações.

- Verificação da estrutura de sua solicitação.

Cada operação do ElastiCache tem uma página de referência na Referência da API do ElastiCache. Verifique novamente se você está usando os parâmetros corretamente. Para conceder ideias sobre o que pode estar errado, consulte as amostras de solicitações ou cenários de usuários para ver se esses exemplos estão realizando operações similares.

- Verificação do fórum.

O ElastiCache tem um fórum de discussão onde você pode procurar soluções para os problemas que outros usuários enfrentaram ao longo do caminho. Para exibir o fórum, consulte

<https://forums.aws.amazon.com/> .

Configuração da interface de linha de comando do ElastiCache

Esta seção descreve os pré-requisitos para executar as ferramentas da linha de comando, em que lugar obtê-las e como configurar as ferramentas e o ambiente, e inclui ainda uma série de exemplos comuns de uso das ferramentas.

Siga as instruções neste tópico somente se for usar a AWS CLI para o ElastiCache.

Important

A CLI (Command Line Interface) do Amazon ElastiCache não oferece suporte a melhorias do ElastiCache após a versão 2014-09-30 da API. Para usar a funcionalidade do ElastiCache mais recente na linha de comando, use a opção [Interface de linha de comando da AWS](#).

Tópicos

- [Pré-requisitos](#)
- [Obtenção das ferramentas da linha de comando](#)
- [Configuração das ferramentas](#)
- [Como fornecer credenciais para as ferramentas](#)
- [Variáveis de ambiente](#)

Pré-requisitos

Este documento pressupõe que você pode trabalhar em um ambiente Linux/UNIX ou Windows. As ferramentas da linha de comando do Amazon ElastiCache também funcionam no Mac OS X, que é um ambiente baseado no UNIX; contudo, não há nenhuma instrução específica ao Mac OS X neste guia.

Como convenção, todos os textos da linha de comando recebem prefixo com um prompt da linha de comando genérico **PROMPT>** . O prompt da linha de comando real da sua máquina pode ser diferente. Também usamos **\$** para indicar um comando específico do Linux/UNIX e **C:\>** para um comando específico do Windows. O exemplo de saída resultante do comando é mostrado imediatamente após, sem qualquer prefixo.

O Java Runtime Environment

As ferramentas da linha de comando usadas neste guia requerem a execução do Java versão 5 ou posterior. Uma instalação do JRE ou do JDK é aceitável. Para ver e fazer download de JREs para uma série de plataformas, incluindo Linux/UNIX e Windows, consulte [Java SE Downloads](#).

Configuração da variável home do Java

As ferramentas da linha de comando dependem de uma variável de ambiente (JAVA_HOME) para localizar o Java Runtime. Essa variável de ambiente deve ser definida para o caminho completo do diretório que contém um subdiretório denominado `bin`, que, por sua vez, contém o executável `java` (no Linux e no UNIX) ou o executável `java.exe` (no Windows).

Para configurar da variável Java Home

1. Configure a variável Java Home.

- No Linux e UNIX, digite o comando a seguir:

```
$ export JAVA_HOME=<PATH>
```

- No Windows, digite o seguinte comando:

```
C:\> set JAVA_HOME=<PATH>
```

2. Confirme a configuração do caminho executando `$JAVA_HOME/bin/java -version` e verificando a saída.

- No Linux/UNIX, você verá uma saída semelhante à seguinte:

```
$ $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)
```

- No Windows, você verá uma saída semelhante à seguinte:

```
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)
```

Obtenção das ferramentas da linha de comando

As ferramentas da linha de comando estão disponíveis como arquivo ZIP no [site Ferramentas de desenvolvedor do ElastiCache](#). Essas ferramentas são escritas em Java e incluem scripts de shell para Windows 2000/XP/Vista/Windows 7, Linux/UNIX e Mac OSX. O arquivo ZIP é independente e nenhuma instalação é necessária; basta fazer download do arquivo ZIP e decompactá-lo em um diretório de seu computador local.

Configuração das ferramentas

As ferramentas da linha de comando dependem de uma variável de ambiente (AWS_ELASTICACHE_HOME) para localizar bibliotecas compatíveis. Você precisa definir essa variável de ambiente para poder usar as ferramentas. Indique o caminho do diretório no qual você descompactou as ferramentas da linha de comando. Esse diretório é chamado ElastiCacheCli-A.B.nnnn (A, B e n são os números de versão/release) e contém os subdiretórios denominados bin e lib.

Para definir a variável de ambiente AWS_ELASTICACHE_HOME

- Abra uma janela de linha de comando e insira um dos seguintes comandos para definir a variável de ambiente AWS_ELASTICACHE_HOME.
 - No Linux e UNIX, digite o comando a seguir:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- No Windows, digite o seguinte comando:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Para facilitar o uso das ferramentas, é recomendável adicionar o diretório BIN das ferramentas ao CAMINHO do sistema. O restante deste guia pressupõe que o diretório BIN encontra-se no caminho do sistema.

Para adicionar o diretório BIN das ferramentas ao caminho do sistema

- Insira os comandos a seguir para adicionar o diretório BIN das ferramentas ao CAMINHO do sistema.

- No Linux e UNIX, digite o comando a seguir:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- No Windows, digite o seguinte comando:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

Note

As variáveis do ambiente Windows são redefinidas quando você fecha a janela de comando. Talvez você queira defini-las permanentemente. Consulte a documentação da versão de seu Windows para obter mais informações.

Note

Os caminhos que contêm um espaço devem ser colocados entre aspas duplas; por exemplo: "C:\Program Files\Java"

Como fornecer credenciais para as ferramentas

As ferramentas da linha de comando precisam da chave de acesso e da chave de acesso secreta da AWS fornecidas com sua conta da AWS. Você pode obtê-las usando a linha de comando ou um arquivo de credenciais localizado no sistema local.

A implantação inclui um arquivo de modelo `${AWS_ELASTICACHE_HOME}/credential-file-path.template` que você precisa alterar com suas informações. A seguir encontra-se o conteúdo do arquivo de modelo:

```
AWSAccessKeyId=<Write your AWS access ID>
AWSSecretKey=<Write your AWS secret key>
```

 Important

No UNIX, limite as permissões ao proprietário do arquivo de credenciais:

```
$ chmod 600 <the file created above>
```

Com o arquivo de credenciais configurado, você precisará definir a variável de ambiente `AWS_CREDENTIAL_FILE` para que as ferramentas do ElastiCache possam encontrar suas informações.

Para definir a variável de ambiente `AWS_CREDENTIAL_FILE`

1. Defina a variável de ambiente:

- No Linux e UNIX, atualize a variável usando o seguinte comando:

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- No Windows, defina a variável usando o seguinte comando:

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. Verifique se sua configuração está funcionando corretamente com o seguinte comando:

```
elasticache --help
```

Você deve examinar a página de uso de todos os comandos do ElastiCache.

Variáveis de ambiente

As variáveis de ambiente podem ser úteis para scripting, configuração de padrões ou para substituí-los temporariamente.

Além da variável de ambiente `AWS_CREDENTIAL_FILE`, a maioria das ferramentas de API incluídas com a interface da linha de comando do ElastiCache comporta as seguintes variáveis:

- `EC2_REGION`: a região da AWS a ser usada.

- **AWS_ELASTICACHE_URL:** o URL a ser usado para a chamada de serviço. Não é necessária para especificar um endpoint regional diferente quando **EC2_REGION** for especificada ou quando se passa o parâmetro `--region`.

Os exemplos a seguir mostram como se define a variável de ambiente **EC2_REGION** para configurar a região usada pelas ferramentas de API:

Linux, OSX ou UNIX

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

Mensagens ElastiCache de erro da Amazon

As seguintes mensagens de erro são retornadas pela Amazon ElastiCache. Você pode receber outras mensagens de erro que são retornadas por ElastiCache outros AWS serviços ou pelo Valkey, Memcached ou Redis OSS. Para obter descrições de mensagens de erro de outras fontes ElastiCache, consulte a documentação da fonte que está gerando a mensagem de erro.

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Manual snapshot quota exceeded](#)
- [Insufficient cluster capacity](#)

Mensagem de erro: Cluster node quota exceeded. Each cluster can have at most %n nodes in this region.

Causa: você tentou criar ou modificar um cluster com o resultado de que o cluster obteve mais de %n nós.

Solução: altere sua solicitação para que o cluster não tenha mais de %n nós. Ou, se você precisar de mais de %n nós, faça sua solicitação usando o [formulário de solicitação Amazon ElastiCache Node](#).

Para obter mais informações, consulte [Amazon ElastiCache Limits](#) in Referência geral da Amazon Web Services.

Mensagens de erro: Customer node quota exceeded. You can have at most %n nodes in this region
Ou, Você já atingiu a cota de %s nós nesta região.

Causa: você tentou criar ou modificar um cluster com o resultado de que sua conta obteve mais de %n nós em todos os clusters nesta região.

Solução: altere sua solicitação para que os nós totais da região em todos os clusters para esta conta não excedam %n. Ou, se você precisar de mais de %n nós, faça sua solicitação usando o [formulário de solicitação Amazon ElastiCache Node](#).

Para obter mais informações, consulte [Amazon ElastiCache Limits](#) in Referência geral da Amazon Web Services.

Mensagens de erro: The maximum number of manual snapshots for this cluster taken within 24 hours has been reached ou The maximum number of manual snapshots for this node taken within 24 hours has been reached its quota of %n

Causa: você tentou tirar um snapshot manual de um cluster quando já tirou o número máximo de snapshots manuais permitidos em um período de 24 horas.

Solução: aguarde 24 horas para tentar outro snapshot manual do cluster. Ou, se precisar tirar um snapshot manual agora, tire o snapshot de outro nó que tenha os mesmos dados, como um nó diferente em um cluster.

Mensagens de erro: InsufficientCacheClusterCapacity

Causa: A AWS não tem capacidade sob demanda disponível suficiente para atender à solicitação no momento.

Solução:

- Espere alguns minutos e envie uma solicitação novamente; a capacidade pode mudar com frequência.

- Envie uma solicitação nova com um número reduzido de nós ou estilhaços (grupos de nós). Por exemplo, se você estiver fazendo uma única solicitação para iniciar 15 nós, tente fazer 3 solicitações de 5 nós, ou 15 solicitações de 1 nó.
- Se você estiver executando um cluster, envie uma nova solicitação sem especificar uma zona de disponibilidade.
- Se você estiver executando um cluster, envie uma nova solicitação usando um tipo de nó diferente (que você pode aumentar posteriormente). Para obter mais informações, consulte [Dimensionamento ElastiCache](#).

Notificações

Este tópico discute as notificações do ElastiCache nas quais você possa estar interessado. Uma notificação é uma situação ou evento que, na maioria dos casos, é temporário, durando apenas até que uma solução seja encontrada e implementada. Em geral, as notificações têm uma data de início e uma data de resolução, após a qual elas deixam de ser relevantes. Qualquer notificação pode ou não ser relevante para você. Recomendamos uma diretriz de implementação que, se seguida, vai melhorar o desempenho do seu cluster.

Notificações não anunciam recursos ou funcionalidades novas ou melhoradas do ElastiCache.

Notificações gerais do ElastiCache

Atualmente não há nenhuma notificação importante do ElastiCache que não seja específica ao mecanismo.

Notificações do ElastiCache for Memcached

As notificações ElastiCache a seguir são específicas ao mecanismo Memcached.

Notificações específicas do ElastiCache for Memcached

- [Alerta: crawler de LRU do Memcached causando falhas de segmentação](#)

Alerta: crawler de LRU do Memcached causando falhas de segmentação

 Data do alerta: 28 de fevereiro de 2017

Em algumas circunstâncias, seu cluster pode exibir instabilidade com uma falha de segmentação no Crawler de LRU do Memcached. Este é um problema no mecanismo Memcached que existe há algum tempo. O problema ficou aparente no Memcached 1.4.33, quando o Crawler de LRU foi habilitado por padrão.

Se você estiver enfrentando esse problema, recomendamos desabilitar o Crawler de LRU até que haja uma correção. Para fazer isso, use `lru_crawler disable` na linha de comando ou modifique o valor do parâmetro `lru_crawler` (preferencial).

Data de resolução:

Resolução:

Notificações específicas do ElastiCache para Redis OSS

Atualmente não há nenhuma notificação importante do ElastiCache para Redis OSS.

ElastiCache Histórico da documentação

- Versão da API: 02/02/2015
- Última atualização da documentação: 27 de novembro de 2023

A tabela a seguir descreve mudanças importantes em cada versão do Guia do ElastiCache usuário após março de 2018. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em o feed RSS.

ElastiCache Atualizações recentes

Alteração	Descrição	Data
Support ElastiCache para escalabilidade vertical sob demanda do Memcached e escalabilidade horizontal automatizada	ElastiCache com o Memcached, agora oferece suporte ao escalonamento vertical sob demanda e ao escalonamento horizontal automatizado. Isso inclui uma atualização da política para a ação <code>elasticache:ModifyCacheCluster</code>	10 de abril de 2025
Support for ElastiCache with Valkey	ElastiCache agora suporta Valkey. O Valkey 7.2.6 é compatível com o Redis OSS 7.2. Para obter mais informações, consulte Valkey .	08 de outubro de 2024
Tamanho de nós reservados flexíveis	ElastiCache agora suporta nós reservados flexíveis de tamanho . Para obter mais informações, consulte Amazon ElastiCache Pricing .	1.º de outubro de 2024

[Support ElastiCache para
Memcached 1.6.22](#)

ElastiCache para Memcached agora oferece suporte ao Memcached 1.6.22. Ele inclui correções de erros cumulativas da versão [1.6.18](#). Para obter mais informações, consulte [Memcached versão 1.6.22](#).

10 de janeiro de 2024

[ElastiCache para Redis OSS
foi adicionado suporte para
tamanhos adicionais de nós
C7gn](#)

ElastiCache para Redis OSS adicionou suporte para tamanhos adicionais de nós C7gn.

10 de janeiro de 2024

[ElastiCache para Redis, o OSS agora suporta a criação de caches sem servidor](#)

Agora você pode criar caches sem servidor, que simplificam o gerenciamento de cache e escalam instantaneamente para dar suporte a aplicações mais exigentes. Para obter mais informações, consulte [Choosing between deployment options](#). Como parte desse recurso, [novas permissões](#) foram adicionadas a `ElastiCacheServiceRolePolicy` e a `AmazonElastiCacheFullAccess` para permitir a associação de caches sem servidor com endpoints da VPC gerenciados. Além disso, as permissões foram adicionadas para dar suporte a uma experiência revisada do console usando a política `AmazonElastiCacheFullAccess`.

27 de novembro de 2023

[ElastiCache para Memcached agora suporta a criação de caches sem servidor](#)

Agora você pode criar caches sem servidor, que simplificam o gerenciamento de cache e escalam instantaneamente para dar suporte a aplicações mais exigentes. Para obter mais informações, consulte [Choosing between deployment options](#). Como parte desse recurso, [novas permissões](#) foram adicionadas a `ElastiCacheServiceRolePolicy` e a `AmazonElastiCacheFullAccess` para permitir a associação de caches sem servidor com endpoints da VPC gerenciados. Além disso, as permissões foram adicionadas para dar suporte a uma experiência revisada do console usando a política `AmazonElastiCacheFullAccess`.

27 de novembro de 2023

[ElastiCache para Redis, o OSS agora suporta a modificação do modo de cluster](#)

Agora você pode migrar clusters do Modo de cluster desativado (CMD) para o Modo de cluster ativado (CME). Para obter mais informações, consulte [Modificar um modo de cluster](#).

11 de maio de 2023

[ElastiCache para Redis, o OSS agora suporta a modificação das configurações de criptografia em trânsito](#)

Agora você pode alterar a configuração de TLS dos seus clusters do Redis OSS sem a necessidade de criar ou provisionar de novo os clusters ou afetar a disponibilidade da aplicação. Para obter mais informações, consulte [Habilitação da criptografia em trânsito em um cluster existente](#).

28 de dezembro de 2022

[ElastiCache para Redis, o OSS agora oferece suporte à autenticação de usuários usando o IAM](#)

A autenticação do IAM permite que você autentique uma conexão com o ElastiCache Redis OSS usando AWS identidades do IAM. Isso possibilita que você fortaleça seu modelo de segurança e simplifique várias tarefas administrativas de segurança. Para obter mais informações, consulte [Autenticação com o IAM](#).

16 de novembro de 2022

[ElastiCache para Redis OSS agora suporta Redis OSS 7](#)

Esta versão traz vários novos recursos para a Amazon ElastiCache para Redis OSS: funções do Redis OSS, melhorias na ACL e Sharded Pub/Sub. Para obter mais informações, consulte [ElastiCache para Redis OSS versão 7.0](#).

8 de novembro de 2022

[ElastiCache para Redis OSS agora suporta IPV6](#)

7 de novembro de 2022

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4 e IPv6), permitindo que você configure seu cluster para aceitar somente IPv4 conexões, somente IPv6 conexões ou ambas IPv4 e IPv6 conexões (pilha dupla). [IPv6 é compatível com cargas de trabalho usando o mecanismo Redis OSS versão 6.2 em diante em todas as instâncias criadas no sistema Nitro.](#) Não há custos adicionais para acessar ElastiCache IPv6. Para obter mais informações, consulte [Escolher um tipo de rede](#).

[ElastiCache para Memcached agora suporta IPV6](#)

7 de novembro de 2022

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4 e IPv6), permitindo que você configure seu cluster para aceitar somente IPv4 conexões, somente IPv6 conexões ou ambas IPv4 e IPv6 conexões (pilha dupla). IPv6 [é compatível com cargas de trabalho usando o mecanismo Memcached versão 1.6.6 em diante em todas as instâncias criadas no sistema Nitro.](#)

Não há custos adicionais para acessar ElastiCache IPv6. Para obter mais informações, consulte [Escolher um tipo de rede.](#)

[ElastiCache para Memcached agora oferece suporte à criptografia em trânsito](#)

26 de maio de 2022

A criptografia em trânsito é um recurso opcional que permite aumentar a segurança de seus dados nos pontos mais vulneráveis quando eles estiverem em trânsito de um local para outro. Ela é compatível com as versões 1.6.12 e posteriores do Memcached. Para obter mais informações, consulte [Criptografia ElastiCache em trânsito \(TLS\).](#)

[ElastiCache para Redis OSS
agora oferece suporte ao
formato nativo de notação de
JavaScript objeto \(JSON\)](#)

O formato nativo de notação de JavaScript objeto (JSON) é uma maneira simples e sem esquemas de codificar conjuntos de dados complexos dentro dos clusters do Redis OSS. Você pode armazenar e acessar dados de forma nativa usando o formato JavaScript Object Notation (JSON) dentro dos clusters Redis OSS e atualizar os dados JSON armazenados nesses clusters, sem precisar gerenciar código personalizado para serializá-los e desserializá-los. Para obter mais informações, consulte [Conceitos básicos do JSON](#).

25 de maio de 2022

[ElastiCache agora suporta
PrivateLink](#)

AWS PrivateLink permite que você acesse de forma privada as operações ElastiCache da API sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão AWS Direct Connect. Para obter mais informações, consulte [Amazon ElastiCache API and interface VPC endpoints \(AWS PrivateLink\)](#) para Redis OSS ou Amazon [API ElastiCache e interface VPC endpoints](#) () para Memcached.AWS PrivateLink

24 de janeiro de 2022

[ElastiCache para Redis OSS agora suporta Redis OSS 6.2 e Data Tiering](#)

O Amazon ElastiCache for Redis OSS apresenta a próxima versão do mecanismo Redis OSS suportado pela Amazon. ElastiCache ElastiCache para Redis, o OSS 6.2 inclui melhorias de desempenho para clusters habilitados para TLS usando tipos de nós x86 com 8 v CPUs ou mais ou tipos de nós Graviton2 com 4 v ou mais. CPUs ElastiCache para Redis, o OSS também introduz o armazenamento de dados em camadas. Você pode usar a classificação de dados em níveis como uma maneira de menor custo para escalar seus clusters para até centenas de terabytes de capacidade. [Para obter mais informações, consulte ElastiCache o Redis OSS versão 6.2 \(aprimorado\) e a classificação por camadas de dados.](#)

23 de novembro de 2021

[Suporte para o Auto Scaling](#)

ElastiCache para Redis, o OSS agora oferece suporte ao Auto Scaling. ElastiCache para Redis OSS, o auto scaling é a capacidade de aumentar ou diminuir automaticamente os fragmentos ou réplicas desejados em seu serviço ElastiCache for Redis OSS. ElastiCache aproveita o serviço Application Auto Scaling para fornecer essa funcionalidade. Para obter mais informações, consulte [Auto Scaling ElastiCache para clusters Redis OSS](#).

19 de agosto de 2021

[Suporte para a entrega de logs lentos do Redis OSS](#)

ElastiCache agora permite que você transmita o Redis OSS SLOWLOG para um dos dois destinos: Amazon Data Firehose ou Amazon Logs. CloudWatch Para obter mais informações, consulte [Entrega de log](#).

22 de abril de 2021

[Suporte à marcação de recursos e chaves de condição](#)

ElastiCache agora oferece suporte à marcação para ajudar você a gerenciar seus clusters e outros ElastiCache recursos. Para obter mais informações, consulte [Como marcar seus ElastiCache recursos](#). ElastiCache também introduz suporte para chaves de condição. Você pode especificar as condições que determinam como uma política do IAM entra em vigor. Para obter mais informações, consulte [Uso de chaves de condição](#).

7 de abril de 2021

[Suporte à marcação de recursos e chaves de condição](#)

ElastiCache agora oferece suporte à marcação para ajudar você a gerenciar seus clusters e outros ElastiCache recursos. Para obter mais informações, consulte [Como marcar seus ElastiCache recursos](#). ElastiCache também introduz suporte para chaves de condição. Você pode especificar as condições que determinam como uma política do IAM entra em vigor. Para obter mais informações, consulte [Uso de chaves de condição](#).

7 de abril de 2021

[ElastiCache agora está disponível em AWS Outposts](#)

[AWS Outposts](#) trazem AWS serviços, infraestrutura e modelos operacionais nativos para praticamente qualquer data center, espaço de co-localização ou instalação local. Você pode implantar ElastiCache no Outposts para configurar, operar e usar o cache no local, da mesma forma que faria na nuvem. Para obter mais informações, consulte [Usar Outposts](#).

8 de outubro de 2020

[ElastiCache agora suporta Redis OSS 6](#)

O Amazon ElastiCache for Redis OSS apresenta a próxima versão do mecanismo Redis OSS suportado pela Amazon. ElastiCache Esta versão inclui [Autenticação de usuários com controle de acesso baseado em função](#), suporte sem versão, cache do lado do cliente e melhorias operacionais significativas. Para obter mais informações, consulte [ElastiCache o Redis OSS versão 6.0 \(aprimorado\)](#).

7 de outubro de 2020

[ElastiCache agora suporta Locais Zones](#)

Uma zona local é uma extensão de uma AWS região geograficamente próxima aos seus usuários. Você pode estender qualquer nuvem privada virtual (VPC) de uma AWS região principal para Zonas Locais criando uma nova sub-rede e atribuindo-a a uma Zona Local. Para obter mais informações, consulte [Uso de Local Zones](#).

25 de setembro de 2020

[ElastiCache para Redis OSS agora oferece suporte à escalabilidade de seu ambiente de cluster Redis OSS para até 500 nós ou 500 fragmentos](#)

O modo cluster do Redis OSS possibilita configurações que você pode usar para particionar seus dados em vários fragmentos e oferece melhor escalabilidade, performance e disponibilidade. Esse recurso está disponível na Amazon ElastiCache for Redis OSS versão 5.0.6 em diante em todas as AWS regiões e para todos os ambientes existentes e novos do Redis OSS ElastiCache Cluster. Para obter mais informações, consulte [Nós e fragmentos do Redis OSS](#).

13 de agosto de 2020

[ElastiCache agora oferece suporte a permissões em nível de recurso](#)

Agora você pode restringir o escopo das permissões de um usuário especificando ElastiCache recursos em uma política AWS Identity and Access Management(IAM). Para obter mais informações, consulte [Permissões no nível do recurso](#).

12 de agosto de 2020

[ElastiCache para Redis OSS adiciona métricas adicionais da Amazon CloudWatch](#)

ElastiCache para Redis, o OSS agora oferece suporte a novas CloudWatch métricas, incluindo e. PubSubCmds HyperLogLogBasedCmds Para obter uma lista completa, consulte [Métricas para Redis OSS](#).

10 de junho de 2020

[ElastiCache agora oferece suporte à atualização automática de clusters ElastiCache](#)

A Amazon ElastiCache agora oferece suporte à atualização automática de ElastiCache clusters após o término da “data de validade recomendada” da atualização do serviço. ElastiCache usará sua janela de manutenção para agendar a atualização automática dos clusters aplicáveis. Para obter mais informações, consulte [Atualizações de autoatendimento](#).

13 de maio de 2020

[ElastiCache para Redis OSS
agora oferece suporte ao
Global Datastore for Redis
OSS](#)

O recurso Global Datastore for Redis OSS oferece replicação totalmente gerenciada, rápida, confiável e segura em todas as regiões. AWS Usando esse recurso, você pode criar clusters de réplicas de leitura entre regiões para o Redis OSS ElastiCache para permitir leituras de baixa latência e recuperação de desastres em todas as regiões. AWS Você pode criar, modificar e descrever um datastore global. Você também pode adicionar ou remover AWS regiões do seu armazenamento de dados global e promover uma AWS região como primária dentro de um armazenamento de dados global. Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global](#).

16 de março de 2020

[ElastiCache para Redis OSS
agora suporta Redis OSS
versão 5.0.6](#)

Para obter mais informações, consulte ElastiCache a [versão 5.0.6 \(aprimorada\) do Redis OSS](#).

18 de dezembro de 2019

[A Amazon ElastiCache agora oferece suporte a nós de cache padrão T3](#)

Agora você pode iniciar a próxima geração de nós de cache T3 padrão intermitentes de uso geral na Amazon ElastiCache. As instâncias T3 Standard EC2 da Amazon fornecem um nível básico de desempenho da CPU com a capacidade de aumentar o uso da CPU a qualquer momento até que os créditos acumulados se esgotem. Para obter mais informações, consulte [Tipos de nó com suporte](#).

12 de novembro de 2019

[A Amazon ElastiCache agora suporta a modificação do token AUTH em um servidor OSS existente ElastiCache para Redis](#)

ElastiCache para Redis, o OSS 5.0.6 agora permite que você modifique os tokens de autenticação configurando e girando novos tokens. Agora é possível modificar tokens ativos enquanto eles estão sendo usados. Também é possível adicionar tokens novos a clusters existentes habilitados com a criptografia em trânsito que foi configurada anteriormente sem tokens de autenticação. Esse é um processo de duas etapas pelo qual é possível definir e alternar o token sem interromper as solicitações de clientes. Atualmente, esse recurso não é suportado no CloudFormation. Para obter mais informações, consulte [Autenticação de usuários com o comando Redis OSS AUTH](#).

30 de outubro de 2019

[A Amazon ElastiCache agora oferece suporte à migração de dados on-line do Redis OSS na Amazon EC2](#)

Agora você pode usar o Online Migration para migrar seus dados do Redis OSS auto-hospedado na Amazon para a Amazon EC2. ElastiCache Para obter mais informações, consulte [Migração online para ElastiCache](#).

28 de outubro de 2019

[ElastiCache for Redis OSS introduz o escalonamento vertical on-line para o modo Redis OSS Cluster.](#)

Agora você pode ampliar ou reduzir seu cluster Redis OSS fragmentado sob demanda. ElastiCache para Redis, o OSS redimensiona seu cluster alterando o tipo de nó, enquanto o cluster continua on-line e atende às solicitações recebidas. Para obter mais informações, consulte [Escalabilidade vertical online modificando o tipo de nó.](#)

20 de agosto de 2019

[ElastiCache O OSS for Redis agora permite que os usuários usem um único endpoint de leitura para seu cluster Amazon ElastiCache for Redis OSS.](#)

Esse recurso permite que você direcione todo o tráfego de leitura ElastiCache para seu cluster do Redis OSS por meio de um único endpoint em nível de cluster para aproveitar o balanceamento de carga e a maior disponibilidade. Para obter mais informações, consulte [Encontrar endpoints de conexão.](#)

13 de junho de 2019

[ElastiCache para Redis, o OSS agora permite que os usuários apliquem atualizações de serviço de acordo com sua própria programação](#)

Com este recurso, você pode optar por aplicar atualizações de serviços disponíveis no horário de sua escolha, e não apenas durante as janelas de manutenção. Isso minimizará as interrupções do serviço, especialmente durante os picos de fluxo de negócios, e ajudará a garantir que você permaneça em conformidade se seu cluster estiver em programas ElastiCache de conformidade compatíveis. Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache](#) e [Validação de conformidade para a Amazon ElastiCache](#).

4 de junho de 2019

[ElastiCache Ofertas de instância reservada padrão: adiantamento parcial, pagamento adiantado total e sem pagamento adiantado.](#)

As instâncias reservadas oferecem a flexibilidade de reservar uma ElastiCache instância da Amazon por um período de um ou três anos com base em um tipo de instância e AWS região. Para obter mais informações, consulte [Gerenciamento de custos com nós reservados](#).

18 de janeiro de 2019

[ElastiCache para suporte do Redis OSS para até 250 nós por cluster do Redis OSS](#)

O limite de nós ou fragmentos pode ser aumentado para um máximo de 250 ElastiCache por cluster Redis OSS. Para obter mais informações, consulte [Fragmentos](#).

19 de novembro de 2018

[ElastiCache para suporte do Redis OSS para failover automático, backup e restauração em todos os nós T2](#)

ElastiCache para Redis O OSS introduz suporte para failover automático, criação de instantâneos e backup e restauração em todos os nós T2. [Para obter mais informações, consulte ElastiCache Backup and Restore and Snapshot do Redis OSS.](#)

19 de novembro de 2018

[ElastiCache para suporte do Redis OSS para nós M5 e R5](#)

ElastiCache para Redis, o OSS agora oferece suporte aos nós M5 e R5, tipos de instância de uso geral e otimizados para memória com base no Sistema Nitro.AWS Para obter mais informações, consulte [Tipos de nó com suporte](#).

23 de outubro de 2018

[Suporte para alteração dinâmica do número de réplicas de leitura](#)

ElastiCache para Redis, o OSS adicionou suporte para adicionar e remover réplicas de leitura de qualquer cluster sem tempo de inatividade do cluster. Para obter mais informações sobre essas e outras mudanças nesta versão, consulte [Alterando o número de réplicas no Guia do usuário do OSS ElastiCache for Redis](#). Veja também [DecreaseReplicaCount](#) e [IncreaseReplicaCount](#) na Referência da ElastiCache API.

17 de setembro de 2018

[Certificação de compatibilidade FedRAMP](#)

ElastiCache para Redis, o OSS agora é certificado para conformidade com o FedRAMP. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

30 de agosto de 2018

[Atualizações do mecanismo do Valkey ou Redis \(modo cluster habilitado\)](#)

O Amazon ElastiCache for Redis OSS adicionou suporte para atualizar as versões do mecanismo Valkey ou Redis OSS (modo de cluster ativado). Para obter mais informações, consulte [Atualizar versões do mecanismo](#).

20 de agosto de 2018

[Certificação de conformidade com PCI DSS](#)

ElastiCache para Redis, o OSS agora é certificado para conformidade com o PCI DSS. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

5 de julho de 2018

[Support ElastiCache para Redis OSS 4.0.10](#)

ElastiCache para Redis OSS agora oferece suporte ao Redis OSS 4.0.10, incluindo criptografia e redimensionamento de clusters online em uma única versão. Para obter mais informações, consulte ElastiCache o [Redis OSS versão 4.0.10](#) (aprimorado).

14 de junho de 2018

[Guia do usuário reestruturado](#)

O [Guia ElastiCache do Usuário único agora está reestruturado para que haja guias de usuário separados para o Redis OSS \(para o Guia do Usuário do Redis OSS\) e ElastiCache para o Memcached \(para o Guia do Usuário do Memcached\)](#). [ElastiCache](#) A estrutura da documentação na seção [AWS CLI Command Reference : elasticache](#) e na [Amazon ElastiCache API Reference](#) permanece inalterada.

20 de abril de 2018

[Support for Engine CPU Utilization metric](#)

ElastiCache for Redis OSS adicionou uma nova métrica, `EngineCPUUtilization`, que relata a porcentagem da capacidade e da sua CPU que está sendo usada atualmente. Para obter mais informações, consulte [Métricas do Redis OSS](#).

9 de abril de 2018

A tabela a seguir descreve as mudanças importantes no Guia do ElastiCache usuário antes de março de 2018.

Alteração	Descrição	Alterado em
Suporte para a região Ásia-Pacífico (Osaka-local).	ElastiCache adicionou suporte para a região Ásia-Pacífico (local de Osaka). Atualmente, a região Ásia-Pacífico (Osaka) oferece suporte a uma única zona de disponibilidade e somente por meio de convite. Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	12 de fevereiro de 2018
Suporte para a Europa (Paris).	ElastiCache adicionou suporte para a região da UE (Paris). Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	18 de dezembro de 2017

Alteração	Descrição	Alterado em
Suporte para a região da China (Ningxia)	A Amazon ElastiCache adicionou suporte para a região da China (Ningxia). Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	11 de dezembro de 2017
Suporte para funções vinculadas ao serviço	Esta versão ElastiCache adicionou suporte para Service Linked Roles (SLR). Para saber mais, consulte: • Uso de funções vinculadas ao serviço para o Amazon ElastiCache • Configure suas permissões (somente para novos ElastiCache usuários)	7 de dezembro de 2017
Suporte para tipos de nós R4	Esta versão ElastiCache adicionou tipos de nós de suporte R4 em todas as AWS regiões suportada s pelo ElastiCache. Você pode adquirir os tipos de nó R4 sob demanda ou como nós de cache reservados. Para saber mais, consulte: • Tipos de nó de cache compatíveis • Parâmetros específicos do tipo de nó do Memcached • Parâmetros específicos de node-type do Redis OSS	20 de novembro de 2017

Alteração	Descrição	Alterado em
ElastiCache para Redis OSS 3.2.10 e suporte para refragmentação online	O Amazon ElastiCache for Redis OSS adiciona suporte ElastiCache para o Redis OSS 3.2.10. ElastiCache para Redis, o OSS também introduz o redimensionamento de clusters on-line para adicionar ou remover fragmentos do cluster enquanto ele continua atendendo às solicitações recebidas. I/O Para saber mais, consulte: • Redimensionamento de cluster on-line • Refragmentação online para o Valkey ou o Redis OSS (modo cluster habilitado)	9 de novembro de 2017
Elegibilidade para HIPAA	ElastiCache para Redis, a versão 3.2.6 do OSS agora é certificada para elegibilidade para HIPAA quando a criptografia está habilitada em seu cluster. Para saber mais, consulte: • Validação de conformidade para a Amazon ElastiCache • Segurança de dados no Amazon ElastiCache	2 de novembro de 2017

Alteração	Descrição	Alterado em
ElastiCache para Redis OSS 3.2.6 e suporte para criptografia	ElastiCache adiciona suporte ElastiCache para o Redis OSS 3.2.6, que inclui dois recursos de criptografia: • A criptografia em trânsito criptografa seus dados sempre que eles estão em trânsito, como entre os nós em um cluster ou entre um cluster e o aplicativo. • A criptografia em repouso criptografa dados no disco durante as operações de sincronização e backup. Para saber mais, consulte: • Segurança de dados no Amazon ElastiCache • Mecanismos e versões compatíveis	25 de outubro de 2017
Tópico de padrões de conexão	ElastiCache a documentação adiciona um tópico que abrange vários padrões de acesso a um ElastiCache cluster em uma Amazon VPC. Consulte mais informações em Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC no Guia de Usuário ElastiCache .	24 de abril de 2017
Suporte para Memcached 1.4.34	ElastiCache adiciona suporte ao Memcached versão 1.4.34, que incorpora várias correções às versões anteriores do Memcached. Para obter mais informações, consulte as notas de lançamento do Memcached 1.4.34 em Memcached on. GitHub	10 de abril de 2017

Alteração	Descrição	Alterado em
Suporte para testar o failover automático	ElastiCache adiciona suporte para testar o failover automático em clusters Redis OSS que oferecem suporte à replicação. Para saber mais, consulte: • Teste do failover automático no ElastiCache Guia do usuário. • TestFailover, na Referência de APIs do ElastiCache . • test-failover na AWS CLI Referência.	4 de abril de 2017
Restauração avançada do Redis OSS	ElastiCache adiciona backup e restauração aprimorados do Redis OSS com redimensionamento de clusters. Esse recurso oferece suporte para a restauração de um backup para um cluster com um número diferente de fragmentos do que o cluster usado para criar o backup. (Para a API e CLI, esse recurso pode restaurar um número diferente de grupos de nó em vez de um número diferente de fragmentos.) Esta atualização também oferece suporte a diferentes configurações de slot do Redis OSS. Para obter mais informações, consulte Restauração de um backup para um novo cache.	15 de março de 2017

Alteração	Descrição	Alterado em
Novo parâmetro Redis OSS de gerenciamento de memória	ElastiCache adiciona um novo parâmetro Redis OSS, <code>reserved-memory-percent</code>, que facilita o gerenciamento da memória reservada. Esse parâmetro está disponível em todas as versões do ElastiCache for Redis OSS. Para saber mais, consulte: • Gerenciamento de memória reservada para Valkey e Redis OSS • Novos parâmetros para o Redis OSS 3.2.4	15 de março de 2017
Suporte para Memcached 1.4.33	ElastiCache adiciona suporte para a versão 1.4.33 do Memcached. Para saber mais, consulte: • ElastiCache versão 1.4.33 para Memcached • Parâmetros adicionados do Memcached 1.4.33	20 de dezembro de 2016
Suporte para a região oeste da Europa (Londres)	ElastiCache adiciona suporte para a região da UE (Londres). No momento, há suporte apenas para os tipos de nós T2 e M4. Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	13 de dezembro de 2016
Suporte à região Canadá (Montreal)	ElastiCache adiciona suporte para a região do Canadá (Montreal). Somente os tipos de nó M4 e T2 são atualmente suportados nesta AWS região. Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	8 de dezembro de 2016

Alteração	Descrição	Alterado em
Suporte para tipos de nós M4 e R3	ElastiCache adiciona suporte aos tipos de nós R3 e M4 na região da América do Sul (São Paulo) e aos tipos de nós M4 na região da China (Pequim). Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	1º de novembro de 2016
Suporte para a região Leste dos EUA 2 (Ohio)	ElastiCache adiciona suporte para a região Leste dos EUA (Ohio) (us-east-2) com os tipos de nós M4, T2 e R3. Para saber mais, consulte as informações a seguir. • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis	17 de outubro de 2016

Alteração	Descrição	Alterado em
Suporte para o cluster do Redis OSS	ElastiCache adiciona suporte ao Redis OSS Cluster (aprimorado). Os clientes que usam o cluster do Redis OSS podem particionar seus dados em até 15 fragmentos (grupos de nós). Cada fragmento oferece suporte para replicação com até 5 réplicas de leitura por fragmento. Os tempos de failover automático do cluster do Redis OSS são cerca de um quarto dos tempos das versões anteriores. Este release inclui um console de gerenciamento reprojetoado que usa terminologia de acordo com o uso da indústria. Para saber mais, consulte: • Comparação entre o Memcached e o Redis OSS • ElastiCache componentes e recursos: observe as seções sobre Nós, Fragmentos, Clusters e Replicação. • ElastiCache terminologia	12 de outubro de 2016

Alteração	Descrição	Alterado em
Suporte para tipo de nós M4	ElastiCache adiciona suporte para a família M4 de tipos de nós na maioria das AWS regiões suportadas peloElastiCache. Você pode adquirir os tipos de nó M4 sob demanda ou como nós de cache reservados. Para saber mais, consulte: • Tipos de nó de cache compatíveis • Parâmetros específicos do tipo de nó do Memcached • Parâmetros específicos de node-type do Redis OSS	3 de agosto de 2016
Suporte para a região de Mumbai	ElastiCache adiciona suporte para a região Ásia-Pacífico (Mumbai). Para saber mais, consulte: • Tipos de nó de cache compatíveis • Parâmetros específicos do tipo de nó do Memcached • Parâmetros específicos de node-type do Redis OSS	27 de junho de 2016
Exportação de snapshots	ElastiCache adiciona a capacidade de exportar um instantâneo do Redis OSS para que você possa acessá-lo de fora. ElastiCache Para saber mais, consulte: • Exportação de um backupno Guia do ElastiCache usuário da Amazon • CopySnapshotna Amazon ElastiCache API Reference	26 de maio de 2016
Ampliação do tipo de nó	ElastiCache adiciona a capacidade de escalar seu tipo de nó do Redis OSS. Para obter mais informações, consulte Dimensionamento ElastiCache.	24 de março de 2016

Alteração	Descrição	Alterado em
Atualização fácil do mecanismo	ElastiCache adiciona a capacidade de atualizar facilmente seu mecanismo de cache Redis OSS. Para obter mais informações, consulte Gerenciamento de versões para ElastiCache .	22 de março de 2016
Suporte para tipos de nós R3	ElastiCache adiciona suporte para tipos de nós R3 na região da China (Pequim) e na região da América do Sul (São Paulo). Para obter mais informações, consulte Supported cache node types .	16 de março de 2016
Acessando ElastiCache usando uma função Lambda	Foi adicionado um tutorial sobre como configurar uma função Lambda para ElastiCache acessar em uma Amazon VPC. Para obter mais informações, consulte Outros tutoriais e vídeos do ElastiCache .	12 de fevereiro de 2016
Suporte para Redis OSS 2.8.24	ElastiCache adiciona suporte para o Redis OSS versão 2.8.24 com melhorias adicionadas desde o Redis OSS 2.8.23. Entre os aprimoramentos incluem correções de bugs e suporte para o registro em log de endereços de acesso à memória incorretos. Para saber mais, consulte: • ElastiCache versão 2.8.24 para Redis OSS (aprimorada) • Redis OSS 2.8 Notas de release	20 de janeiro de 2016
Suporte para a região Ásia-Pacífico (Seul).	ElastiCache adiciona suporte para a região Ásia-Pacífico (Seul) (ap-northeast-2) com os tipos de nós t2, m3 e r3.	6 de janeiro de 2016

Alteração	Descrição	Alterado em
Mudança ElastiCache no console da Amazon.	Como as versões mais recentes do Redis OSS oferecem uma experiência de usuário melhor e mais estável, as versões 2.6.13, 2.8.6 e 2.8.19 do Redis OSS não estão mais listadas no Management Console. ElastiCache Para outras opções e mais informações, consulte Mecanismos e versões compatíveis .	15 de dezembro de 2015
Suporte para Redis OSS 2.8.23.	ElastiCache adiciona suporte para o Redis OSS versão 2.8.23 com melhorias adicionadas desde o Redis OSS 2.8.22. Entre os aprimoramentos estão correções de bugs e suporte para o novo parâmetro <code>close-on-slave-write</code> que, se ativado, desconecta clientes que tentam gravar em uma réplica somente leitura. Para obter mais informações, consulte ElastiCache versão 2.8.23 para Redis OSS (aprimorada) .	13 de novembro de 2015

Alteração	Descrição	Alterado em
Suporte para Redis OSS 2.8.22.	ElastiCache adiciona suporte para o Redis OSS versão 2.8.22 com aprimoramentos e melhorias ElastiCache adicionais desde a versão 2.8.21. Entre os aprimoramentos estão: • Implementação de um processo de salvamento sem bifurcação que permite um salvamento bem-sucedido quando a memória disponível baixa pode fazer com que uma gravação com bifurcação falhe. • CloudWatch Métricas adicionais — SaveInProgressReplicationBytes. • Para habilitar sincronizações parciais, o parâmetro <code>repl-backlog-size</code> do Redis OSS agora se aplica a todos os clusters. Para obter uma lista completa de mudanças e mais informações, consulte ElastiCache versão 2.8.22 para Redis OSS (aprimorada). Esta versão da documentação inclui uma reorganização da documentação e a remoção da documentação da interface de linha de comando (CLI). Para usar a linha de comando, consulte a Linha de AWS comando para ElastiCache.	28 de setembro de 2015

Alteração	Descrição	Alterado em
Suporte para o Memcached 1.4.28.	ElastiCache adiciona suporte para o Memcached versão 1.4.24 e melhorias no Memcached desde a versão 1.4.14. Esta versão adiciona suporte para gerenciamento de cache LRU (menos utilizado recentemente) como uma tarefa em segundo plano, a opção por jenkins ou murmur3 como algoritmo de hashing, novos comandos e correções de diversos erros. Para obter mais informações, as Notas de release do Memcached .	27 de agosto de 2015
Suporte para a Descoberta automática do Memcached usando PHP 5.6	Esta versão da Amazon ElastiCache adiciona suporte ao cliente Memcached Auto Discovery para PHP versão 5.6. Para obter mais informações, consulte Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP .	29 de julho de 2015
Suporte para Redis OSS 2.8.21	ElastiCache adiciona suporte para o Redis OSS versão 2.8.21 e melhorias no Redis OSS desde a versão 2.8.19. Esta versão do Redis OSS inclui diversas correções de bugs. Para obter mais informações, consulte as Notas de release do Redis OSS 2.8 .	29 de julho de 2015
Novo tópico: Acessando ElastiCache de fora AWS	Foi adicionado um novo tópico sobre como acessar ElastiCache recursos de fora AWS. Para obter mais informações, consulte Acessando ElastiCache recursos de fora AWS .	9 de julho de 2015

Alteração	Descrição	Alterado em
Mensagens de substituição de nó adicionadas	ElastiCache adiciona três mensagens relacionadas à substituição programada do nó, ElastiCache: NodeReplacementScheduled, ElastiCache: NodeReplacementRescheduled, ElastiCache: NodeReplacementCanceled. Para obter mais informações e ações que você pode tomar quando um nó está programado para ser substituído, consulte ElastiCache Notificações de eventos e o Amazon SNS.	11 de junho de 2015

Alteração	Descrição	Alterado em
Suporte para Redis OSS v.2.8.19.	ElastiCache adiciona suporte para o Redis OSS versão 2.8.19 e melhorias no Redis OSS desde a versão 2.8.6. Este suporte inclui suporte para: • A estrutura HyperLogLog de dados, com os comandos do Redis OSS PFADD, PFCOUNT e PFMERGE. • Consultas de alcance lexicográfico com os novos comandos ZRANGEBYLEX, ZLEXCOUNT e ZREMRANGEBYLEX. • Uma série de correções de erros, impedindo que um nó primário envie dados obsoletos aos nós de réplica, falhando o SYNC primário quando um processo filho de salvamento em segundo plano (bgsave) é encerrado inesperadamente. Para obter mais informações sobre HyperLogLog, consulte a nova estrutura de dados do Redis OSS: a. HyperLogLog Para obter mais informações sobre PFADD, PFCOUNT e PFMERGE, consulte a documentação do Redis OSS e clique em. HyperLogLog	11 de março de 2015
Suporte para tags de alocação de custos	ElastiCache adiciona suporte para tags de alocação de custos. Para obter mais informações, consulte Monitoramento de custos com tags de alocação de custos.	9 de fevereiro de 2015
Support para a região AWS GovCloud (Oeste dos EUA)	ElastiCache adiciona suporte para a região AWS GovCloud (Oeste dos EUA) (us-gov-west-1).	29 de janeiro de 2015

Alteração	Descrição	Alterado em
Suporte para a região Europa (Frankfurt)	ElastiCache adiciona suporte para a região Europa (Frankfurt) (eu-central-1).	19 de janeiro de 2015
Suporte a Multi-AZ para grupos de replicação do Redis OSS	ElastiCache adiciona suporte para Multi-AZ do nó primário a uma réplica de leitura em um grupo de replicação do Redis OSS. ElastiCache monitora a integridade do grupo de replicação. Se o primário falhar, promove ElastiCache automaticamente uma réplica para primária e, em seguida, substitui a réplica. Para obter mais informações, consulte Minimizar o tempo de inatividade no ElastiCache usando o Multi-AZ com Valkey e Redis OSS .	24 de outubro de 2014
AWS CloudTrail registro de chamadas de API suportado	ElastiCache adiciona suporte para uso AWS CloudTrail para registrar todas as chamadas de ElastiCache API. Para obter mais informações, consulte Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail .	15 de setembro de 2014
Novos tamanhos de instâncias com suporte	ElastiCache adiciona suporte para instâncias adicionais de uso geral (T2). Para obter mais informações, consulte Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache .	11 de setembro de 2014
Posicionamento flexível de nós para Memcached	ElastiCache adiciona suporte para criar nós do Memcached em várias zonas de disponibilidade.	23 de julho de 2014

Alteração	Descrição	Alterado em
Novos tamanhos de instâncias com suporte	ElastiCache adiciona suporte para instâncias adicionais de uso geral (M3) e instâncias otimizadas para memória (R3). Para obter mais informações, consulte Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache .	1º de julho de 2014
Descoberta automática PHP	Suporte adicionado para Descoberta automática do PHP versão 5.5.	13 de maio de 2014
Backup e restauração para clusters do Redis OSS	Nesta versão, ElastiCache permite que os clientes criem instantâneos de seus clusters Redis OSS e criem novos clusters usando esses instantâneos. Um backup é uma cópia do cluster em um momento específico e consiste em metadados do cluster e todos os dados no cache do Redis OSS. Os backups são armazenados no Amazon S3, e os clientes podem restaurar os dados de um snapshot para um novo cluster a qualquer momento. Para obter mais informações, consulte Snapshots e restauração .	24 de abril de 2014
Redis OSS 2.8.6	ElastiCache suporta o Redis OSS 2.8.6, além do Redis OSS 2.6.13. Com o Redis OSS 2.8.6, os clientes podem melhorar a resiliência e a tolerância a falhas de réplicas de leitura, com suporte para ressincronização parcial e um número mínimo definido pelo usuário de réplicas de leitura que devem estar disponíveis em todos os momentos. O Redis OSS 2.8.6 também oferece suporte completo para publish-and-subscribe, onde os clientes podem ser notificados sobre eventos que ocorrem no servidor.	13 de março de 2014

Alteração	Descrição	Alterado em
Mecanismo de cache do Redis OSS	ElastiCache oferece o software de mecanismo de cache Redis OSS, além do Memcached. Os clientes que atualmente usam o Redis OSS podem “semear” um novo cluster do ElastiCache Redis OSS com seus dados existentes a partir de um arquivo de instantâneo do Redis OSS, facilitando a migração para um ambiente gerenciado. ElastiCache Para oferecer suporte aos recursos de replicação do Redis OSS, a ElastiCache API agora oferece suporte a grupos de replicação. Os clientes podem criar um grupo de replicação com um nó de cache Redis OSS primário e adicionar um ou mais nós de réplica de leitura que permanecem automaticamente sincronizados com dados de cache no nó primário. Aplicativos que exigem leitura intensa podem ser descarregados para uma réplica de leitura, reduzindo a carga no nó primário. Réplicas de leitura também podem proteger contra a perda de dados no caso de uma falha do nó de cache primário.	3 de setembro de 2013
Suporte para a Amazon Virtual Private Cloud (VPC) padrão.	Nesta versão, ElastiCache está totalmente integrado com a Amazon Virtual Private Cloud (VPC). Para novos clientes, os clusters são criados em uma Amazon VPC por padrão. Para obter mais informações, consulte Amazon VPCs e ElastiCache segurança .	8 de janeiro de 2013
Suporte a PHP para Descoberta automática de nós de cache	O release inicial da Descoberta automática de nós de cache fornecia suporte para programas Java. Nesta versão, ElastiCache traz suporte à descoberta automática de nós de cache para PHP.	2 de janeiro de 2013

Alteração	Descrição	Alterado em
Suporte para a Amazon Virtual Private Cloud (VPC).	Nesta versão, os ElastiCache clusters podem ser lançados na Amazon Virtual Private Cloud (VPC). Por padrão, clusters de novos clientes são criados em uma Amazon VPC automaticamente, enquanto os clientes existentes podem migrar para a Amazon VPC ao seu próprio ritmo. Para obter mais informações, consulte Amazon VPCs e ElastiCache segurança .	20 de dezembro de 2012
Descoberta automática de nós de cache e nova versão do mecanismo de cache	ElastiCache fornece descoberta automática de nós de cache — a capacidade dos programas clientes determinarem automaticamente todos os nós de cache em um cluster e iniciarem e manterem conexões com todos esses nós. Esta versão também oferece uma nova versão de mecanismo de cache: Memcached versão 1.4.14. Esse novo mecanismo de cache oferece uma melhor capacidade de rebalanceamento de slabs, melhorias significativas de desempenho e escalabilidade e várias correções de bugs. Existem vários novos parâmetros de cache que podem ser configurados. Para obter mais informações, consulte Configuração de parâmetros do mecanismo usando grupos de parâmetros do ElastiCache.	28 de novembro de 2012
Novos tipos de nó de cache	Esse release fornece quatro tipos de nó de cache adicionais.	13 de novembro de 2012
Nós de cache reservados	Esse release adiciona suporte para nós de cache reservados.	5 de abril de 2012
Novo guia	Esta é a primeira versão do Amazon ElastiCache User Guide.	22 de agosto de 2011

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Glossário da AWS Referência.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.