



Guia do usuário

# Amazon ElastiCache



Versão da API 2015-02-02

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

# Amazon ElastiCache: Guia do usuário

Copyright © 2024 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 ElastiCache é .....	1
Armazenamento em cache sem servidor .....	1
Clusters autoprojetoados .....	2
Serviços relacionados .....	2
Como funciona .....	3
Cache e mecanismos de cache .....	3
Escolha entre opções de implantação .....	9
ElastiCache recursos .....	16
AWS Regiões e zonas de disponibilidade .....	18
Casos de uso .....	20
Começando com ElastiCache .....	28
Configurando ElastiCache .....	28
Inscreva-se para um Conta da AWS .....	28
Criar um usuário com acesso administrativo .....	29
Conceder acesso programático .....	30
Configurar permissões .....	32
Configurar EC2 .....	33
Conceder acesso de rede .....	34
Configurar o acesso à linha de comando .....	35
Crie um cache sem servidor Valkey .....	36
Ler e gravar dados .....	37
Limpeza .....	39
Próximos Passos .....	40
Crie um cache sem servidor Valkey ou Redis OSS .....	40
Ler e gravar dados .....	41
Limpeza .....	43
Próximos Passos .....	44
Crie um cache sem servidor do Memcached .....	44
Ler e gravar dados .....	46
Limpeza .....	50
Próximos Passos .....	51
Tutoriais: Introdução ao Python e ElastiCache .....	51
Python e ElastiCache .....	52
Tutorial: Configurando o Lambda para acessar em um ElastiCache VPC .....	70

Etapa 1: Criar um cache ElastiCache sem servidor .....	70
Etapa 2: criar uma função Lambda para ElastiCache .....	73
Etapa 3: Teste a função Lambda com ElastiCache .....	77
Etapa 4: limpar (opcional) .....	77
Projetando seu próprio ElastiCache cluster .....	79
Componentes e recursos .....	79
Nodes .....	80
ElastiCache cacos .....	81
ElastiCache aglomerados .....	81
ElastiCache replicação .....	83
ElastiCache endpoints .....	86
Grupos de parâmetros .....	87
ElastiCache segurança .....	87
Grupos de sub-redes .....	88
ElastiCache cópias de segurança .....	88
Eventos .....	88
ElastiCache terminologia .....	90
Tutorial: Como criar seu próprio cluster .....	92
Projetando seu próprio ElastiCache cluster (Valkey) .....	92
Projetando seu próprio ElastiCache cluster (RedisOSS) .....	114
Excluir um cluster .....	136
Outros tutoriais e vídeos .....	138
Vídeos .....	139
Gerenciando nós em ElastiCache .....	144
Visualizando o status ElastiCache do nó .....	145
OSSNódulos e fragmentos Valkey ou Redis .....	150
Conexão a nós .....	153
Tipos de nó compatíveis .....	158
Reinicializando nós .....	173
Substituindo nós (Valkey e RedisOSS) .....	178
Substituindo nós (Memcached) .....	185
Nós reservados .....	186
Migração de nós da geração anterior .....	203
Gerenciando clusters em ElastiCache .....	206
Escolhendo um tipo de rede em ElastiCache .....	209
Descoberta automática (Memcached) .....	214



Hierarquização de dados em ElastiCache .....	256
Preparando um cluster em ElastiCache .....	263
Criação de um cluster para Valkey ou Redis OSS .....	274
Criação de um cluster para Memcached .....	284
Visualizando os detalhes de um ElastiCache cluster .....	288
Modificando um cluster ElastiCache .....	302
Adicionando nós a um ElastiCache cluster .....	308
Removendo nós de um ElastiCache cluster .....	319
Cancelamento de operações pendentes de adição ou exclusão de nós no ElastiCache .....	328
Excluindo um cluster no ElastiCache .....	329
Acessando seu ElastiCache cluster ou grupo de replicação .....	332
Encontrando pontos de extremidade de conexão em ElastiCache .....	341
Fragmentos em ElastiCache .....	355
Comparando os caches autoprotetidos do Valkey, do Redis OSS e do Memcached .....	361
Migração on-line para Valkey ou Redis OSS .....	367
Visão geral .....	367
Etapas da migração .....	368
Preparando sua origem e destino para a migração .....	368
Testar a migração de dados .....	370
Início da migração .....	371
Verificação do andamento do processo de migração de dados .....	372
Conclusão da migração de dados .....	373
Execução de migração de dados online usando o console .....	374
Escolhendo regiões e zonas de disponibilidade para ElastiCache .....	375
Considerações sobre a zona de disponibilidade com o Memcached .....	376
Localização dos seus nós .....	379
Regiões e endpoints com suporte .....	379
Usando zonas locais com ElastiCache .....	384
Usando Outposts com ElastiCache .....	386
Trabalhando com ElastiCache .....	391
Snapshots e restauração .....	391
Restrições .....	392
Impacto sobre o desempenho dos backups de clusters autoprotetidos .....	393
Programação de backups automáticos .....	395
Realização de backups manuais .....	397
Criação de um backup final .....	403

Descrição de backups .....	406
Copiar backups .....	408
Exportação de um backup .....	410
Restauração de um backup .....	418
Exclusão de um backup .....	421
Marcação de backups .....	422
Tutorial: semeando um cluster autoprojetoado com um backup .....	423
Versões e atualizações do motor em ElastiCache .....	433
Gerenciamento de versões para ElastiCache .....	433
Como atualizar as versões dos mecanismos .....	438
Como acionar a atualização entre mecanismos do Redis OSS para o Valkey .....	439
Versões compatíveis .....	440
Principais diferenças de comportamento e compatibilidade da versão com o Valkey .....	463
Principais diferenças de comportamento e compatibilidade da versão com o Redis OSS ....	463
Resolvendo atualizações de motor bloqueadas com Valkey ou Redis OSS .....	467
Práticas recomendadas e estratégias de armazenamento em cache .....	468
Melhores práticas gerais .....	468
Comandos Valkey, Redis e Memcached suportados OSS e restritos .....	470
OSSConfiguração e limites do Valkey e do Redis .....	507
IPv6exemplos de clientes para Valkey, Redis OSS e Memcached .....	510
Melhores práticas para clientes (Valkey e RedisOSS) .....	511
Melhores práticas para clientes (Memcached) .....	535
TLSclusters de pilha ElastiCache dupla habilitados .....	538
Gerenciando a memória reservada para Valkey e Redis OSS .....	542
Práticas recomendadas ao trabalhar com clusters autoprojetoados do Valkey e do Redis OSS .....	549
Estratégias de armazenamento em cache para Memcached .....	555
Gerenciando seu cluster autoprojetoado em ElastiCache .....	560
Clusters Auto Scaling Valkey e Redis OSS .....	561
Modificar o modo de cluster .....	610
Replicação entre AWS regiões usando datastores globais .....	613
Alta disponibilidade com o uso de grupos de replicação .....	642
Gerenciando a manutenção ElastiCache do cluster .....	733
Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros .....	736
Dimensionamento ElastiCache .....	844
Dimensionamento sem servidor ElastiCache .....	844

Definição dos limites de escalabilidade para gerenciar custos .....	845
Pré-escalonamento com Serverless ElastiCache .....	845
Definindo limites de escala usando o console e AWS CLI .....	846
Dimensionamento de clusters autoprojetoados .....	848
Introdução ao JSON Valkey e ao Redis OSS .....	924
JSONvisão geral do tipo de dados .....	925
JSONcomandos .....	937
Marcando seus recursos ElastiCache .....	979
Monitoramento de custos com tags .....	991
Gerenciando tags usando o AWS CLI .....	992
Gerenciando tags usando o ElastiCache API .....	997
Lente Amazon ElastiCache Well-Architected .....	1000
Pilar Excelência operacional .....	1001
Pilar de segurança .....	1010
Pilar Confiabilidade .....	1016
Pilar Eficiência de performance .....	1023
Pilar da otimização de custos .....	1034
Solução de problemas em ElastiCache .....	1041
Problemas de conectividade .....	1041
Erros do cliente Valkey ou Redis OSS .....	1042
Solução de problemas de alta latência no Serverless ElastiCache .....	1043
Solução de problemas de limitação no Serverless ElastiCache .....	1045
Problemas persistentes de conexão .....	1046
Related Topics .....	1067
Segurança .....	1068
Proteção de dados .....	1069
Segurança de dados na Amazon ElastiCache .....	1069
Privacidade do tráfego entre redes .....	1148
Amazon VPCs e ElastiCache segurança .....	1148
ElastiCache API e VPC endpoints de interface ( )AWS PrivateLink .....	1174
Sub-redes e grupos de sub-redes .....	1178
Identity and Access Management .....	1186
Público .....	1186
Autenticando com identidades .....	1187
Gerenciando acesso usando políticas .....	1191
Como a Amazon ElastiCache trabalha com IAM .....	1193

Exemplos de políticas baseadas em identidade .....	1200
Solução de problemas .....	1203
Controle de acesso .....	1205
Visão geral do gerenciamento de acesso .....	1206
Validação de conformidade .....	1253
Mais informações .....	1255
Resiliência .....	1255
Atenuar falhas .....	1256
Segurança da infraestrutura .....	1261
Atualizações de serviço .....	1261
Gerenciando atualizações de serviços .....	1262
Vulnerabilidades de segurança abordadas .....	1267
Logging e monitoramento .....	1270
Métricas e eventos sem servidor para Valkey e Redis OSS .....	1270
Métricas sem servidor .....	1270
Eventos sem servidor .....	1281
Métricas e eventos de cluster projetados por você mesmo .....	1294
Métricas e eventos sem servidor para Memcached .....	1304
Métricas sem servidor .....	1304
Eventos sem servidor .....	1308
Registrando ElastiCache API chamadas da Amazon com AWS CloudTrail .....	1318
ElastiCache Informações da Amazon em CloudTrail .....	1319
Entendendo as entradas do arquivo de ElastiCache log da Amazon .....	1320
Monitoramento SNS de eventos da Amazon .....	1323
Gerenciando SNS notificações ElastiCache da Amazon .....	1324
Visualizando ElastiCache eventos .....	1329
Notificações de eventos e Amazon SNS .....	1333
Entrega de logs .....	1340
Conteúdo de uma entrada de log lento .....	1341
Conteúdo de uma entrada de log do mecanismo .....	1342
Permissões para configurar os logs .....	1342
Especificações de tipo de log e formato de log .....	1343
ElastiCache destinos de registro .....	1344
Especificação da entrega de logs usando o console .....	1347
Especificando a entrega de registros usando o AWS CLI .....	1348
Monitoramento do uso .....	1353

Métricas em nível de host .....	1354
Métricas para Valkey e Redis OSS .....	1357
Métricas para o Memcached .....	1375
Que métricas devo monitorar? .....	1381
Escolher estatísticas e períodos de métricas .....	1385
Monitorando métricas de CloudWatch clusters e nós .....	1385
Cotas .....	1389
Referência .....	1391
Usando o ElastiCache API .....	1391
Usando a consulta API .....	1391
Bibliotecas disponíveis .....	1395
Solução de problemas de aplicações .....	1395
Configure o AWS CLI para ElastiCache .....	1396
Pré-requisitos .....	1397
Obtenção das ferramentas da linha de comando .....	1399
Configuração das ferramentas .....	1399
Como fornecer credenciais para as ferramentas .....	1400
Variáveis de ambiente .....	1401
Mensagens de erro .....	1402
Notificações .....	1404
ElastiCache Notificações gerais .....	1404
ElastiCache Notificações (Memcached) .....	1404
ElastiCache Notificações específicas (RedisOSS) .....	1405
ElastiCache Histórico da documentação .....	1406
AWS Glossário .....	1446
.....	mcdxlvii

# 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 sem servidor ou optar por criar seu próprio cluster de cache.

## Note

A Amazon ElastiCache trabalha com os mecanismos Valkey, Redis OSS e Memcached. Se não souber qual mecanismo deseja usar, consulte [Comparando os caches autoprojetados do Valkey, do Redis OSS e do Memcached](#) 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 ou SDK 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, Redis OSS 7.1 e superior e Memcached 1.6.21 e superior.

# Projetando seu próprio ElastiCache cluster

Se precisar de um controle refinado sobre seu ElastiCache cluster, você pode optar por criar seu próprio cluster Valkey, Redis OSS ou Memcached com. ElastiCache permite que você projete seu cluster, 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.

Projetar seu próprio ElastiCache cluster oferece maior flexibilidade e controle sobre seus 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, o Redis OSS ou o Memcached no modo de cluster, permitindo o escalonamento horizontal, ou sem o modo de cluster, para escalar apenas verticalmente. Ao projetar os próprios clusters, 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 seu cluster Valkey ou RedisOSS.

Ao projetar seu próprio ElastiCache cluster, você pode optar por executar o Valkey 7.2, o Redis OSS 4.0 a 7.1 ou o Memcached 1.4 e superior.

## Serviços relacionados

### [Memória DB](#)

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

- ElastiCache é um serviço comumente usado para armazenar dados em cache de outros bancos de dados e armazenamentos de dados usando Valkey, Redis OSS ou Memcached. Você deve considerar ElastiCache armazenar em cache cargas de trabalho onde deseja acelerar o acesso aos dados com seu banco de dados primário ou armazenamento de dados existente (desempenho de leitura e gravação em microssegundos). Você também deve considerar os casos ElastiCache de uso em que deseja usar estruturas de dados Valkey ou Redis e APIs acessar OSS dados armazenados em um banco de dados ou armazenamento de dados primário.
- ElastiCache também pode ajudá-lo a economizar custos de banco de dados armazenando dados acessados com frequência em um cache. Se seu aplicativo tiver altos requisitos de taxa de transferência de leitura, você poderá obter alta escala, desempenho rápido e custos reduzidos

de armazenamento de dados usando, em vez de escalar ElastiCache, seu banco de dados subjacente.

- O MemoryDB é um banco de dados durável em memória para cargas de trabalho que exigem um banco de dados primário ultrarrápido. É compatível com Valkey e RedisOSS. 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 um aplicativo usando estruturas de dados Valkey ou Redis e APIs com um banco de OSS dados primário 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](#)

ElastiCache pode ajudá-lo a economizar custos de banco de dados armazenando dados acessados com frequência em um cache. Se seu aplicativo tiver altos requisitos de taxa de transferência de leitura, você poderá obter alta escala, desempenho rápido e custos reduzidos de armazenamento de dados usando, em vez de escalar ElastiCache, seu banco de dados subjacente.

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

ElastiCache pode ajudá-lo a economizar custos de banco de dados armazenando dados acessados com frequência em um cache. Se seu aplicativo tiver altos requisitos de taxa de transferência de leitura, você poderá obter alta escala, desempenho rápido e custos reduzidos de armazenamento de dados usando, em vez de escalar ElastiCache, seu 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 você pode usar para armazenar dados em cache. Normalmente, a aplicação vai armazenar em cache os dados mais acessados em um cache para otimizar os tempos de resposta. ElastiCache oferece duas opções de implantação: clusters sem servidor e com design próprio. Consulte [Escolha entre opções de implantação](#).



**Note**

A Amazon ElastiCache trabalha com os mecanismos Valkey, Redis OSS e Memcached. Se não souber qual mecanismo deseja usar, consulte [Comparando os caches autoprotetidos do Valkey, do Redis OSS e do Memcached](#) neste guia.

## Tópicos

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

## 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. Você simplesmente fornece um nome para seu cache e recebe um único endpoint que pode ser configurado em seu cliente Valkey, Redis OSS ou Memcached para começar a acessar seu cache.

**Note**

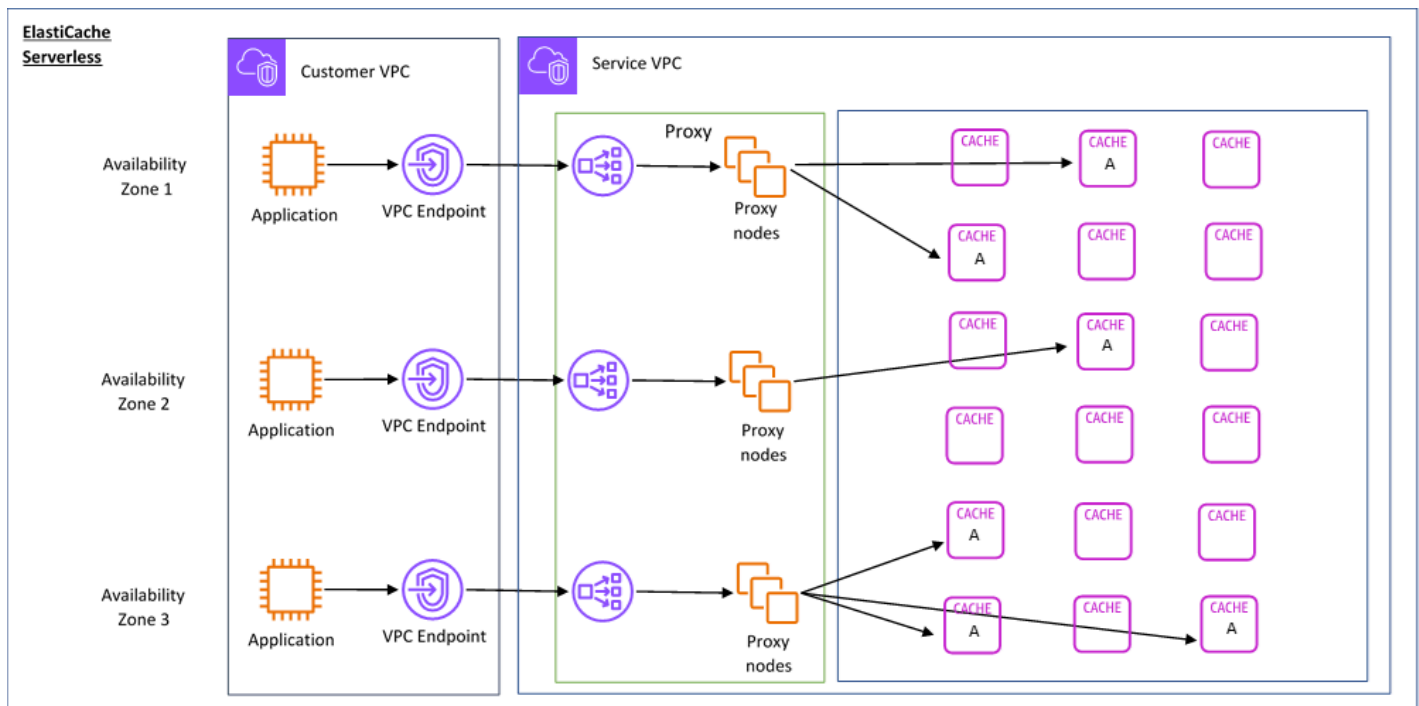
- ElastiCache O Serverless executa Valkey, Redis OSS ou Memcached no modo de cluster e só é compatível com clientes que oferecem suporte. TLS

## 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 uma disponibilidade de 99,99% SLA 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 no seu VPC. Seu aplicativo pode se conectar ao cache por meio desses VPC endpoints.

Com o ElastiCache Serverless, você recebe um único DNS endpoint 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 de cache 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 autoprojettados ElastiCache

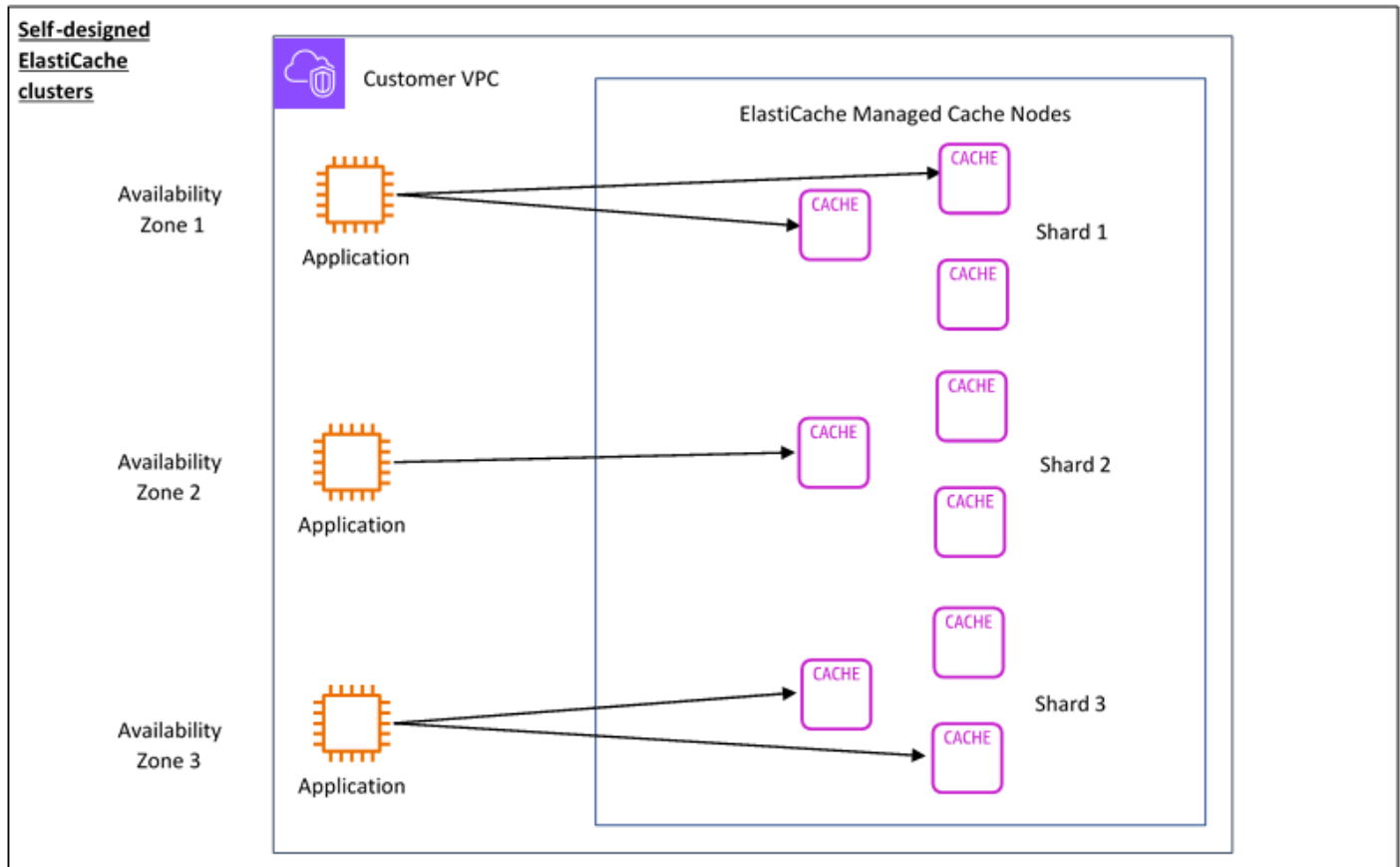
Você pode escolher criar seus próprios ElastiCache clusters escolhendo uma família de nós de cache, tamanho e número de nós para seu cluster. Projetar o próprio cluster proporciona um controle mais refinado e permite a você escolher o número de fragmentos no cache e o número de nós (primário e réplica)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

## Benefícios principais

- **Crie seu próprio cluster:** com ElastiCache, você pode criar seu próprio cluster 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 projetar seu cluster com nós em vários AZs para obter alta disponibilidade.
- **Controle refinado:** ao projetar o próprio cluster, você tem mais controle sobre o ajuste fino das configurações no cache. Por exemplo, você pode usar [Parâmetros Valkey e Redis OSS](#) ou [Parâmetros específicos do Memcached](#) 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. Você também pode usar o recurso Auto-Scaling para configurar o escalonamento com base em

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

O diagrama a seguir ilustra como os clusters ElastiCache autoprojeto 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 Unidades de Processamento (UPUs). ElastiCache ECPU Ao escolher criar seus próprios ElastiCache clusters, 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 CPU tempo v e os dados transferidos.

- Leituras e gravações simples exigem 1 ECPU para cada quilobyte (KB) de dados transferidos. Por exemplo, um GET comando que transfere até 1 KB de dados consome 1 ECPU. Uma SET solicitação que transfere 3,2 KB de dados consumirá 3,2 ECPUs.
- Com o Valkey e o RedisOSS, os comandos que consomem mais CPU tempo v e transferem mais dados consomem ECPUs com base na maior das duas dimensões. Por exemplo, se seu aplicativo usa o HMGET comando, consome 3 vezes o CPU tempo v do que um simples GET comando SET/ 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 RedisOSS, os comandos que exigem CPU tempo v adicional consumirão proporcionalmente mais. ECPUs Por exemplo, se seu aplicativo usa o OSS [HMGET comando](#) Valkey ou Redis e consome 3 vezes o CPU tempo v do que um simples GET comando SET/, 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 seu aplicativo usar o GET comando, recuperar 3 itens e transferir 3,2 KB de dados, ele 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 escolher criar seu próprio cluster de cache escolhendo a família de EC2 nós, o tamanho, o número de nós e o posicionamento nas zonas de disponibilidade. Ao projetar o cluster por conta própria, você paga por hora por nó de cache.

## ElastiCache cópias de segurança

Um backup é uma point-in-time cópia de um cache sem servidor ou de um cluster autoprotetido pelo Valkey ou RedisOSS. 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 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 autoprotetidos

Para obter uma lista dos comandos compatíveis com ambos, consulte [Comandos Valkey, Redis e Memcached suportados OSS 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 de cache. ElastiCache O Serverless armazena automaticamente os dados de forma redundante em três zonas de disponibilidade e fornece um Acordo de Nível de Serviço de 99,99% de disponibilidade (). SLA Os backups de OSS clusters Valkey ou Redis autoprotetidos podem ser restaurados em uma configuração sem servidor.

### Clusters autoprotetidos

Se você precisar de um controle refinado sobre seu cluster Valkey, Redis ou MemcachedOSS, você pode optar por criar seu próprio cluster com. ElastiCache 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 ajuda a gerenciar o provisionamento de hardware, o monitoramento, as substituições de nós e a aplicação de patches de software para seu cluster. Clusters autoprotetidos podem ser projetados para fornecer uma disponibilidade de até 99,99%. SLA Os backups de OSS caches Valkey ou Redis sem servidor podem ser restaurados em um cluster autoprotetido.

### Escolha entre opções de implantação

Escolha o armazenamento sem servidor se:

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

Escolha criar seu próprio ElastiCache cluster se:

- Você já está executando o ElastiCache Serverless e quer um controle mais refinado sobre o tipo de nó que executa Valkey, Redis ou MemcachedOSS, 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.

Comparando o armazenamento em cache sem servidor e os clusters autoprojeados

Atributo	Armazenamento em cache sem servidor	Clusters autoprojeados
Configuração de cache	Crie um cache com apenas um nome em menos de um minuto	Fornecer controle refinado sobre o design do cluster de cache. 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 de cluster (Valkey e RedisOSS)	Opera <code>cluster mode enabled</code> apenas os motores. Os clientes devem oferecer suporte <code>cluster mode</code>	Pode ser configurado para operar no modo cluster ativado ou no modo cluster desativado.

Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
	enabled para se conectar ao ElastiCache Serverless.	
Escalabilidade	Dimensiona automaticamente os motores vertical e horizontalmente sem nenhum gerenciamento de capacidade.	<p>Fornece controle sobre o dimensionamento, além de exigir monitoramento para garantir que a capacidade atual atenda adequadamente à demanda.</p> <p>Para Valkey e RedisOSS, você pode optar por escalar verticalmente aumentando ou diminuindo o tamanho do nó de cache quando necessário. Você também pode escalar horizontalmente adicionando novos fragmentos ou adicionando mais réplicas aos seus fragmentos. Esse recurso não está disponível para o Memcached.</p> <p>Com o recurso Auto-Scaling, você também pode configurar o escalonamento com base em uma programação ou escalar com base em métricas como o uso CPU de memória no cache.</p>



Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
Conexão do cliente	Os clientes se conectam a um único endpoint. Isso permite que a topologia do nó de cache subjacente (escalabilidade, 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 redescobrirá a topologia do cluster e restabelecerá as conexões.
Configurabilidade	Nenhuma configuração refinada disponível. Os clientes podem definir 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 autoprotetidos 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 <a href="#">Parâmetros específicos do motor</a> .
Multi-AZ	Os dados são replicados de forma assíncrona em várias zonas de disponibilidade para maior disponibilidade e maior latência de leitura.	Fornece uma opção para projetar o cluster em uma única zona de disponibilidade ou em várias zonas de disponibilidade (AZs). Ao usar Valkey ou RedisOSS, fornece clusters Multi-AZ com dados replicados de forma assíncrona em várias zonas de disponibilidade para maior disponibilidade e maior latência de leitura.

Atributo	Armazenamento em cache sem servidor	Clusters autoprojeto
Criptografia em repouso	Sempre ativado. Os clientes podem usar uma chave gerenciada pela AWS ou uma chave gerenciada pelo cliente AWS KMS.	Opção para ativar ou desativar a criptografia em repouso. Quando ativada, os clientes podem usar uma chave gerenciada pela AWS ou uma chave gerenciada pelo cliente AWS KMS.
Criptografia em trânsito (TLS)	Sempre ativado. Os clientes devem oferecer suporte à TLS conectividade.	Opção para ativar ou desativar .
Backups	<p>Oferece suporte a backups automáticos e manuais de caches sem impacto no desempenho.</p> <p>Os OSS backups do Valkey e do Redis são compatíveis entre si e podem ser restaurados em um cache ElastiCache sem servidor ou em um cluster autoprojeto.</p>	<p>Suporta backups automáticos e manuais para Valkey e RedisOSS. Os clusters podem ter algum impacto no desempenho, dependendo da memória reservada disponível. Para obter mais informações, consulte <a href="#">Gerenciando a memória reservada para Valkey e Redis OSS</a>.</p> <p>Os OSS backups do Valkey e do Redis são compatíveis entre si e podem ser restaurados em um cache ElastiCache sem servidor ou em um cluster autoprojeto.</p>

Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
Monitorar	<p>Support métricas em nível de cache, incluindo taxa de acerto do cache, taxa de erro do cache, tamanho dos dados e ECPUs consumo.</p> <p>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 <a href="#">Eventos de cache sem servidor</a>.</p>	<p>ElastiCache clusters autoprotetidos emitem métricas em cada nível de nó, incluindo métricas em nível de host e métricas de cache.</p> <p>Clusters autoprotetidos emitem SNS notificações para eventos significativos. Consulte <a href="#">Métricas para o Memcached</a> e <a href="#">Métricas para Valkey e Redis OSS</a>.</p>
Disponibilidade	<p><a href="#">Contrato de nível de serviço de disponibilidade de 99,99%</a> ( ) SLA</p>	<p>Clusters autoprotetidos podem ser projetados para alcançar um <a href="#">Acordo de Nível de Serviço (SLA) de disponibilidade de</a> até 99,99%, dependendo da configuração.</p>

Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
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 da versão principal e podem fazer o upgrade para a versão principal mais recente quando quiserem.	Clusters autoprotetidos oferecem autoatendimento habilitado pelo cliente para atualizações de versões menores e com patches, bem como atualizações de versões principais. As atualizações gerenciadas são aplicadas automaticamente durante os períodos de manutenção definidos pelo cliente. Os clientes também podem optar por aplicar um upgrade de versão secundária ou de patch sob demanda.
Armazenamento de dados global	Sem compatibilidade	Suporta o Global Data Store, que permite a replicação entre regiões com gravações em uma única região e leituras em várias regiões

Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
Hierarquização de dados	Sem compatibilidade	Os clusters projetados usando nós da família r6gd têm seus dados em camadas entre a memória e o armazenamento local SSD (unidades de estado sólido). O armazenamento de dados em camadas fornece uma opção de preço-desempenho para OSS cargas de trabalho Valkey e Redis, 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 ElastiCache de processamento (vCPU). Consulte detalhes de preço <a href="#">aqui</a> .	Pay-per-hour, com base no uso do nó de cache. Consulte detalhes de preço <a href="#">aqui</a> .

Tópicos relacionados:

- [Projetando e gerenciando seu próprio ElastiCache cluster](#)

## ElastiCache Recursos da Amazon para usuários iniciantes

Recomendamos que os usuários iniciantes comecem lendo as seções a seguir e consulte-as 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 [ElastiCache Vídeos](#) 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 cluster de cache. Ele também inclui como autorizar o acesso ao cluster de cache, conectar-se a um nó de cache e excluir o cluster de 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.

- [ElastiCache melhores práticas e estratégias de armazenamento em cache](#)

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 o 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 codificar 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.

- [Usando o ElastiCache API](#)

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

- [ElastiCache APIReferência da Amazon](#)

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

## AWS Regiões e zonas de disponibilidade

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 do data center é chamada de AWS região.

Cada AWS região contém vários locais distintos chamados de zonas de disponibilidade, ouAZs. Cada zona de disponibilidade é projetada para ser isolada de falhas em outras zonas de disponibilidade. Cada uma foi projetada para fornecer conectividade de rede barata e de baixa latência a outras zonas de disponibilidade na mesma região. AWS Ao iniciar as instâncias em zonas de disponibilidade separadas, você pode proteger seus aplicativos de falhas 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 [Minimizando o tempo de inatividade no ElastiCache \(RedisOSS\)](#) 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? Até mesmo a consulta de banco de dados ou API chamada remota mais otimizada é visivelmente mais lenta do que recuperar uma chave simples de um cache na 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](#)
- [Tabelas de classificação de jogos](#)
- [Mensagens \(Pub/Sub\)](#)
- [Dados de recomendação \(hashes\)](#)
- [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.

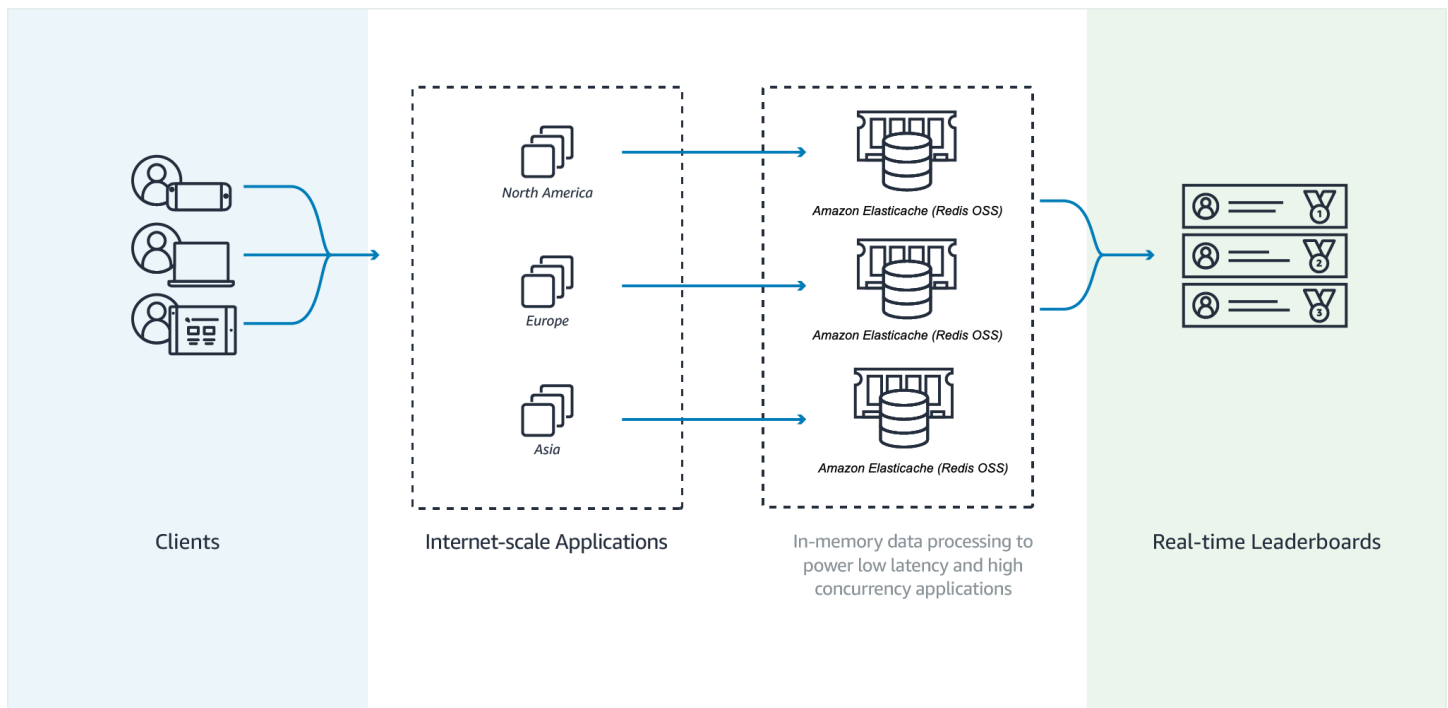
Para ter mais informações, consulte [Estratégias de armazenamento em cache para Memcached](#)

## Tabelas de classificação de jogos

Com os conjuntos OSS classificados do Valkey ou do Redis, você pode mover a complexidade computacional das tabelas de classificação do seu aplicativo para o 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 OSS classificados Valkey e Redis garantem a exclusividade e a ordenação dos elementos. Com conjuntos ordenados, cada vez que um novo elemento é adicionado ao conjunto ordenado, ele é reclassificado 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 Tabela de classificação 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 a classificação é baseada em zero, ZREVRANK retorna 1 para junho, que está na segunda posição.

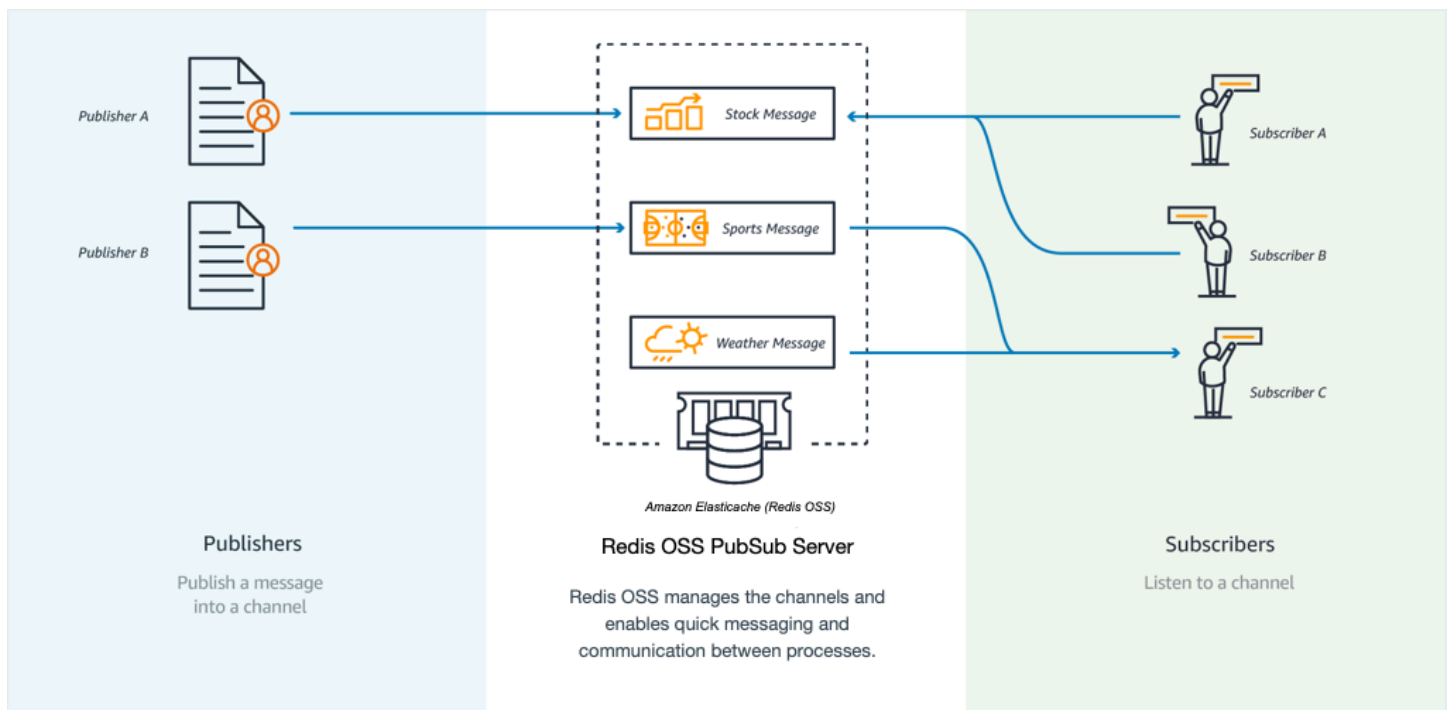
```
ZREVRANK leaderboard June
1
```

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

## Mensagens (Pub/Sub)

Ao enviar uma mensagem de e-mail, você a envia para um ou mais destinatários especificados. No paradigma OSS pub/sub do Valkey e do Redis, você envia uma mensagem para um canal específico sem saber quem, se é que alguém, a recebe. 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 do Pub/Sub não tem relação com nenhum espaço de teclas. Portanto, ela não interferirá em nenhum nível. No diagrama a seguir, você pode encontrar uma ilustração de ElastiCache mensagens com Valkey e RedisOSS.



## 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 se inscrever em um único canal, use o SUBSCRIBE comando que especifica o canal no qual você deseja se inscrever. 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 UNSUBSCRIBE comando que especifica o canal do qual cancelar a assinatura.

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

### Example - Assinaturas de vários canais especificados

Para se inscrever em vários canais específicos, liste os canais com o SUBSCRIBE comando. 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 a assinatura de um canal específico, use o UNSUBSCRIBE comando e especifique o canal do qual cancelar a assinatura.

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

Para cancelar assinaturas em vários canais, use o UNSUBSCRIBE comando 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
```

ou

```
UNSUBSCRIBE
```

### Example - Assinaturas usando a correspondência de padrões

Os clientes podem se inscrever em todos os canais que correspondam a um padrão usando o PSUBSCRIBE comando.

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 cadeia de caracteres do canal enviada para um SUBSCRIBE comando [P] e para o UNSUBSCRIBE comando [P] deve corresponder. Você não pode PSUBSCRIBE em news.\* e PUNSUBSCRIBE de news.sports.\* ou UNSUBSCRIBE de news.sports.golf.

### 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 [Pub/Sub](#) na documentação do Valkey.

## Dados de recomendação (hashes)

Usar INCR ou DECR no Valkey ou no Redis OSS simplifica a compilação das 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 que gostaram ou não de 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
```

## ElastiCache Depoimentos de clientes

Para saber como empresas como AirbnbPBS, 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

- [Configurando ElastiCache](#)
- [Crie um cache sem servidor Valkey](#)
- [Crie um cache sem servidor Valkey ou Redis OSS](#)
- [Crie um cache sem servidor do Memcached](#)
- [Tutoriais: Introdução ao Python e ElastiCache](#)
- [Tutorial: Configurando o Lambda para acessar em um ElastiCache VPC](#)

## Configurando ElastiCache

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](#)
- [Conceda acesso à rede de um grupo VPC de segurança da Amazon ao seu cache](#)
- [Baixe e configure 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 e inserir um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e atributos 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 um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, 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 [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira sua 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. Ative a autenticação multifator (MFA) para seu usuário root.

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

### Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No IAM Identity Center, conceda 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 AWS IAM Identity Center do usuário.

### Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

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 IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

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

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

## Conceder acesso programático

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do AWS Management Console. 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
Identificação da força de trabalho	Use credenciais temporárias para assinar solicitações programáticas para o AWS	Siga as instruções da interface que deseja utilizar.

Qual usuário precisa de acesso programático?	Para	Por
(Usuários gerenciados no IAM Identity Center)	CLI AWS SDKs, ou. AWS APIs	<ul style="list-style-type: none"><li>• Para o AWS CLI, consulte <a href="#">Configurando o AWS CLI para uso AWS IAM Identity Center</a> no Guia do AWS Command Line Interface usuário.</li><li>• Para AWS SDKs, ferramentas e AWS APIs, consulte <a href="#">Autenticação do IAM Identity Center</a> no Guia de referência de ferramentas AWS SDKs e ferramentas.</li></ul>
IAM	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções em <a href="#">Uso de credenciais temporárias com AWS recursos</a> no Guia do IAM usuário.

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. <ul style="list-style-type: none"><li>• Para o AWS CLI, consulte <a href="#">Autenticação usando credenciais de IAM usuário</a> no Guia do AWS Command Line Interface usuário.</li><li>• Para ferramentas AWS SDKs e ferramentas, consulte <a href="#">Autenticar usando credenciais de longo prazo</a> no Guia de referência de ferramentas AWS SDKs e ferramentas.</li><li>• Para AWS APIs, consulte <a href="#">Gerenciamento de chaves de acesso para IAM usuários</a> no Guia IAM do usuário.</li></ul>

Tópicos relacionados:

- [O que está IAM](#) no Guia IAM do usuário.
- [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 AWS IAM Identity Center:

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

- Usuários gerenciados IAM por meio de um provedor de identidade:

Crie um perfil para a federação de identidades. Siga as instruções em [Criação de uma função para um provedor de identidade terceirizado \(federação\)](#) no Guia IAM do usuário.

- IAMusuários:

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

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, verifique se você tem permissões para chamar `iam:createServiceLinkedRole` ou se criou a função ElastiCache vinculada ao serviço.

Para obter mais informações, consulte as informações a seguir.

- [Criando uma nova política](#) (IAM)
- [AWS políticas gerenciadas para a Amazon ElastiCache](#)
- [Usando funções vinculadas a serviços para a Amazon ElastiCache](#)

## Configurar EC2

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

- Se você ainda não tem uma EC2 instância, saiba como configurá-la aqui: [Primeiros passos com EC2](#). EC2

- 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 em seu padrão VPC e usa o grupo de segurança padrão. Para seguir este tutorial, certifique-se de que sua EC2 instância esteja no padrão VPC e tenha o grupo de segurança padrão.

## Conceda acesso à rede de um grupo VPC de segurança da Amazon ao seu cache

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

ElastiCache (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 [EC2console da Amazon](#).
2. No painel de navegação, em Network & Security, escolha Security Groups.
3. Na lista de grupos de segurança, escolha o grupo de segurança para sua AmazonVPC. 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. Selecione a opção Editar.
  - b. Escolha Adicionar regra.
  - c. Na coluna Tipo, escolha TCPRegra personalizada.
  - d. Se estiver usando Valkey ou RedisOSS, 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 Amazon VPC possa se conectar ao seu cache.
  - f. Se você estiver usando a tecnologia ElastiCache sem servidor, adicione outra regra escolhendo Adicionar regra.
  - g. Na coluna Tipo, escolha TCP Regra personalizada.

- h. Se estiver usando ElastiCache (RedisOSS), na caixa Intervalo de portas, digite 6380.

Se estiver usando ElastiCache (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 Amazon VPC possa se conectar ao seu cache.
- j. Escolha Salvar

## Baixe e configure o acesso à linha de comando

Baixe e instale o utilitário valkey-cli.

Se você usa ElastiCache com o Valkey, talvez ache útil o utilitário valkey-cli. Se você estiver usando ElastiCache (RedisOSS) com redis-cli, considere mudar para valkey-cli, pois também funciona para 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 apropriado para sua configuração.

### Amazon Linux 2023

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

### 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/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make BUILD_TLS=yes
```





```
--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 a criação do cache ElastiCache foi concluída, use o `describe-serverless-caches` comando.

## 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 [Leia e grave dados no cache](#).

## Leia e grave 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 configurado o VPC acesso e as configurações do grupo de segurança para a EC2 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 [Configurando ElastiCache](#).

Encontre o endpoint do cache

### AWS Management Console

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

1. Faça login no AWS Management Console 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 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
```

## Conecte-se ao seu Valkey Cache (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"
```

## Conecte-se ao seu Valkey Cache (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 OSS diretório Valkey ou Redis 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 você não precisar mais do ElastiCache cache da Amazon que você criou, você pode excluí-lo. Esta etapa ajuda a garantir que você não será cobrado pelos recursos que não está utilizando. Você pode usar o ElastiCache console AWS CLI, o ou o ElastiCache API para excluir seu cache.

### AWS Management Console

Para excluir o cache usando o console:

1. Faça login no AWS Management Console 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 Valkey Caches.
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 seu cache passa para o DELETINGstatus, você deixa de incorrer em cobranças por ele.

### AWS CLI

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

### Linux

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

## Windows

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

Observe que o valor do campo Status está definido como DELETING.

Agora você já pode avançar para [Próximos Passos](#).

## Próximos Passos

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

- [Trabalhando com ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Registro e monitoramento na Amazon ElastiCache](#)
- [ElastiCache melhores práticas e estratégias de armazenamento em cache](#)
- [Snapshots e restauração](#)
- [SNSMonitoramento de ElastiCache eventos pela Amazon](#)

## Crie um cache sem servidor Valkey ou Redis OSS

Nesta etapa, você cria um novo cache na Amazon ElastiCache.

### AWS Management Console

Para criar um novo cache usando o ElastiCache console:

1. Faça login no AWS Management Console e abra <https://console.aws.amazon.com/connect/o>.
2. No painel de navegação no lado esquerdo do console, escolha caches Valkey ou caches Redis OSS
3. No lado direito do console, escolha Criar cache Valkey ou Criar cache 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.

- Quando o cache estiver no status ACTIVE "", você poderá começar a gravar e ler dados no cache.

## AWS CLI

O AWS CLI exemplo 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 a criação do cache ElastiCache foi concluída, use o describe-serverless-caches comando.

## 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 [Leia e grave dados no cache](#).

## Leia e grave 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 configurado o VPC acesso e as configurações do grupo de segurança para a EC2 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 [Configurando ElastiCache](#).

Encontre o endpoint do cache

## AWS Management Console

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

1. Faça login no AWS Management Console 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 Valkey caches 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
```

## Conecte-se ao seu Valkey ou Redis OSS Cache (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"
```

## Conecte-se ao seu Valkey ou Redis OSS Cache (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, vá para o diretório 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 você não precisar mais do ElastiCache cache da Amazon que você criou, você pode excluí-lo. Esta etapa ajuda a garantir que você não será cobrado pelos recursos que não está utilizando. Você pode usar o ElastiCache console AWS CLI, o ou o ElastiCache API para excluir seu cache.

### AWS Management Console

Para excluir o cache usando o console:

1. Faça login no AWS Management Console 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 Valkey ou OSS Redis Caches.
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 seu cache passa para o DELETINGstatus, você deixa de incorrer em cobranças por ele.

## AWS CLI

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

## Linux

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

## Windows

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

Observe que o valor do campo Status está definido como DELETING.

Agora você já pode avançar para [Próximos Passos](#).

## Próximos Passos

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

- [Trabalhando com ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Registro e monitoramento na Amazon ElastiCache](#)
- [ElastiCache melhores práticas e estratégias de armazenamento em cache](#)
- [Snapshots e restauração](#)
- [SNSMonitoramento de ElastiCache eventos pela Amazon](#)

## Crie um cache sem servidor do Memcached

### AWS Management Console

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

1. Faça login no AWS Management Console e abra o ElastiCache console 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 no status ACTIVE "", você poderá começar a gravar e ler dados no cache.

Para criar um novo cache usando o AWS CLI

O AWS CLI exemplo 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 a criação do cache ElastiCache foi concluída, use o describe-serverless-caches comando.

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 [Leia e grave dados no cache](#).

## Leia e grave 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 em seu padrãoVPC. Certifique-se de que sua EC2 instância também seja criada no padrãoVPC, para que ela possa se conectar ao cache.

Encontre o endpoint do cache

### AWS Management Console

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

1. Faça login no AWS Management Console 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
```

## Conecte-se usando o Open SSL

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

## Conectar usando o cliente Java do Memcached

Para obter informações sobre como se conectar usando o cliente Java do Memcached, consulte [ElastiCache criptografia em trânsito \(\) TLS](#)

## Conecte-se usando o cliente Memcached PHP

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

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

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

## Conectar usando o cliente Python do Memcached (Pymemcache)

Consulte [https://pymemcache.readthedocs.io/en/latest/getting\\_started.html](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"
```

Conectar usando o cliente NodeJS/TS do Memcached (Memcache Electrode-IO)

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

Instalar por meio de npm `i memcache-client`

No aplicativo, crie um TLS cliente memcached da seguinte forma:

```
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 Memcached. NETcliente (EnyimMemcachedCore)

Veja <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 AWS Management Console

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 AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache console, escolha o mecanismo que o cache que você deseja excluir está em execução. 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 seu cache passa para o DELETINGstatus, você deixa de incorrer em 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 delete-serverless-cache CLI ação 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 [Próximos Passos](#).

## Próximos Passos

Para obter mais informações sobre, ElastiCache consulte:

- [Trabalhando com ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Cotas para ElastiCache](#)
- [ElastiCache melhores práticas e estratégias de armazenamento em cache](#)
- [Visualizando ElastiCache eventos](#)

## Tutoriais: Introdução ao Python e ElastiCache

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

### Note

AWS SDKs são disponíveis para uma grande variedade de idiomas. 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 for Python (Boto3) para escrever programas simples para realizar as seguintes operações ElastiCache (Redis): OSS

- Crie clusters ElastiCache (RedisOSS) (modo de cluster ativado e modo de cluster desativado)
- Verifique se existem usuários ou grupos de usuários, caso contrário, crie-os. (Esse recurso está disponível com o Valkey 7.2 e versões posteriores e com o OSS Redis 6.0 em diante.)
- Conecte-se a ElastiCache
- Executar operações como configuração e obtenção de strings, leitura e escrita para streams e publicação e inscrição a partir do canal Pub/Sub.

Ao trabalhar neste tutorial, você pode consultar a documentação do AWS SDK for 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 [Configurando ElastiCache](#).
- 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 boto3 SDK para operações de gerenciamento ElastiCache (RedisOSS) (criação de clusters ou usuários) e o redis-py-cluster redis-py/ para tratamento de dados.

## Tópicos

- [Criar um cluster com modo cluster desabilitado](#)
- [Crie um cluster desativado no modo de cluster com TLS e RBAC](#)
- [Criar um cluster com modo cluster habilitado](#)
- [Crie um cluster habilitado para o modo de cluster com TLS e RBAC](#)
- [Verifique se os usuários/grupo de usuários existem, 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,
  cache_cluster', ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
    cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
    CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
    used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
    node) and 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 7.2, one primary and two replicas
    elasticacheResponse = create_cluster_mode_disabled(
        #CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        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 [Gerenciando clusters em ElastiCache](#).

Crie um cluster desativado no modo de cluster com TLS e RBAC

Para garantir a segurança, você pode usar Transport Layer Security (TLS) e Role-Based Access Control (RBAC) ao criar um cluster desativado no modo de cluster. Ao contrário do Valkey ou do Redis OSSAUTH, em que todos os clientes autenticados têm acesso total ao grupo de replicação se o token for autenticado, RBAC permite que você controle 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 [Controle de acesso baseado em funções \(\) 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
  cache cluster', ReplicationGroupId=None, UserGroupIds=None,
  SecurityGroupIds=None, CacheSubnetGroupName=None):
  """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

  Returns a dictionary with the API response

  :param CacheNodeType: Node type used on the cluster. If not specified,
  cache.t3.small will be used
  Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/CacheNodes.SupportedTypes.html for supported node types
  :param EngineVersion: Engine version to be used. If not specified, latest will be
  used.
  :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
  node) and 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 7.2, 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='7.2',
        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 [Gerenciando clusters em 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 [Gerenciando clusters em ElastiCache](#).

Crie um cluster habilitado para o modo de cluster com TLS e RBAC

Para garantir a segurança, você pode usar Transport Layer Security (TLS) e Role-Based Access Control (RBAC) ao criar um cluster habilitado para o modo de cluster. Ao contrário do Valkey ou do Redis OSSAUTH, em que todos os clientes autenticados têm acesso total ao grupo de replicação se o token for autenticado, RBAC permite que você controle 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 [Controle de acesso baseado em funções \(\) 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 [Gerenciando clusters em ElastiCache](#).

Verifique se os usuários/grupo de usuários existem, caso contrário, crie-os

ComRBAC, 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 ElastiCache (RedisOSS). Ao fazer isso, você pode estabelecer limites de segurança entre clientes usando o mesmo grupo ou grupos de OSS replicação Valkey ou Redis e impedir que os clientes acessem os dados uns dos outros. Para obter mais informações, consulte [Controle de acesso baseado em funções \(\) 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 cadeias 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 OSS cliente Valkey ou Redis para se conectar. 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 boto3 SDK ElastiCache para trabalhar com ElastiCache (RedisOSS).

### Tópicos

- [Definir e obter strings](#)
- [Definir e obter um hash com vários itens](#)
- [Publicar \(escrever\) e assinar \(ler\) a partir de um canal Pub/Sub](#)
- [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
```

## Publicar (escrever) e assinar (ler) a partir de um canal Pub/Sub

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 writing 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 acknowledges 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 acessar em um ElastiCache 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 sem servidor, siga estas etapas.

### Etapa 1.1: Criar um cache sem servidor

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

```
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 de cache sem servidor

Verifique se ElastiCache (RedisOSS) terminou de criar o cache com o `describe-serverless-caches` comando.

```
aws elasticache describe-serverless-caches \  
  --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 IAM função

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

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
    "Action": "sts:AssumeRole"
  },
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "lambda.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

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

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

### 4. Crie a IAM política.

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

### 5. Anexe a IAM política 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 cache sem servidor

### 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 IAM habilitado.

```
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 Lambda

Neste tutorial, fornecemos um exemplo de código em Python para sua função 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`. Certifique-se de substituir o `elasticache_endpoint` valor 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 = boto3.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.us-east-1.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 Python redis-py para colocar itens em seu cache e recuperá-los. Esse código usa ferramentas de cache para armazenar em cache os tokens de IAM autenticação gerados por 15 minutos. Para criar um pacote de implantação contendo redis-py e cachetools, execute as etapas a seguir.

No diretório do projeto contendo o arquivo de código-fonte app.py, crie um pacote de pastas para 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 seu utilitário zip preferido para criar um arquivo.zip com as bibliotecas redis-py e 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 seu utilitário zip preferido para adicionar app.py à raiz do seu arquivo.zip.

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

## Etapa 2.2: Criar a IAM função (função 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 CLI create-function Lambda.

Para a opção de função, use a função ARN de execução que você criou na etapa anterior. Para o vpc-config, insira listas separadas por vírgulas das sub-redes padrão e VPC o ID do grupo de segurança padrão. VPC Você pode encontrar esses valores no VPC console da Amazon. Para encontrar as sub-redes padrãoVPC, escolha Seu e, em seguidaVPCs, escolha o padrão da sua AWS conta. VPC Para encontrar o grupo de segurança para issoVPC, 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 Lambda manualmente usando o comando `invoke`. 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 (`AccessValkey/aws/lambda/`). 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 Lambda

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

### Etapa 4.2: Excluir o cache sem servidor

Exclua o cache.

```
aws elasticache delete-serverless-cache \  

```

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

Remova 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 IAM função e políticas

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

# Projetando e gerenciando seu próprio ElastiCache cluster

Se você precisar de um controle refinado sobre seu ElastiCache cluster, poderá optar por criar seu próprio cluster. 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 [Configurando ElastiCache](#). Para obter detalhes sobre como gerenciar, atualizar ou excluir nós ou clusters, consulte [Gerenciando nós em ElastiCache](#). Para uma visão geral dos principais componentes de uma ElastiCache implantação da Amazon ao projetar seu próprio ElastiCache cluster, consulte esses [conceitos-chave](#).

## Tópicos

- [ElastiCache componentes e recursos](#)
- [ElastiCache terminologia](#)
- [Tutorial: Como criar seu próprio cluster](#)
- [Excluir um cluster](#)
- [Outros ElastiCache tutoriais e vídeos](#)
- [Gerenciando nós em ElastiCache](#)
- [Gerenciando clusters em ElastiCache](#)
- [Comparando os caches autoprojeto do Valkey, do Redis OSS e do Memcached](#)
- [Migração on-line para Valkey ou Redis OSS](#)
- [Escolhendo regiões e zonas de disponibilidade para 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 cópias de segurança](#)
- [ElastiCache eventos](#)

## ElastiCache nós

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

Um nó é um fragmento de tamanho fixo conectado à rede. RAM 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 tem seu próprio nome e porta do Domain Name Service (DNS). 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 [Gerenciando nós em ElastiCache](#).

## ElastiCache cacos

Um OSS fragmento Valkey ou Redis (chamado de grupo de nós no API eCLI) é um agrupamento de um a seis nós relacionados. Um OSS cluster Valkey ou Redis com o modo de cluster ativado sempre tem pelo menos um fragmento.

A fragmentação é um método de particionamento de banco de dados que separa bancos de dados grandes 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.

OSSOs clusters Valkey e Redis com o modo de cluster ativado podem ter até 500 fragmentos, com seus dados particionados entre os fragmentos. O limite de nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do OSS mecanismo Valkey ou Redis 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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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 estilhaços, consulte [Trabalhando com fragmentos em 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 que tenha o modo de OSS cluster ativado.

Muitas ElastiCache operações são direcionadas a clusters:

- Criar 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:

- [Gerenciando clusters em ElastiCache](#) e [Gerenciando nós em 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 de seus clusters e grupos de OSS replicação Valkey ou Redis.

## Configurações de cluster típicas

Veja a seguir configurações de cluster típicas.

### Clusters Valkey ou Redis OSS

OSSOs clusters Valkey ou Redis com o modo de cluster desativado sempre contêm apenas um fragmento (no API eCLI, um grupo de nós). Um OSS fragmento Valkey ou Redis 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 primário de leitura/gravação e os outros são nós de réplica somente de leitura.

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

Conforme a demanda do seu OSS cluster Valkey ou Redis muda, você pode aumentar ou diminuir a escala. Para fazer isso, mova seu cluster para um tipo de instância de nó diferente. Se seu aplicativo exige muita leitura, recomendamos adicionar réplicas somente para 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 [Hierarquização de dados em ElastiCache](#).

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

### 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 Memcached, os clusters podem ser compostos 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 Valkey e RedisOSS, a replicação é implementada agrupando de dois a seis nós em um fragmento (no API eCLI, 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 com Valkey e RedisOSS, e não para ElastiCache (Memcached).

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](#).

OSSOs clusters Valkey ou Redis oferecem suporte a um fragmento (no API eCLI, chamado de grupo de nós).

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

Comparando a replicação: Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado) -> cluster Valkey ou Redis com modo de cluster ativado versus OSS cluster Valkey ou Redis com modo de cluster desativado OSS

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

	Cluster Valkey ou Redis com o modo de OSS cluster desativado	OSSCluster Valkey ou Redis com o modo de cluster ativado
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

	Cluster Valkey ou Redis com o modo de OSS cluster desativado	OSSCluster Valkey ou Redis com o modo de cluster ativado
Suporte a aumento de escala na horizontal	Sim	Sim
Suporte a atualizações do mecanismo	Sim	Sim
Promove réplica para primário	Sim	Automatic
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 backup e restauração para migrar para Valkey ou Redis OSS (modo de cluster ativado).

Você pode usar backup e restauração para redimensionar seu cluster Valkey ou Redis OSS (modo de cluster ativado).

Todos os fragmentos (nos grupos de nós API e CLI,) e nós devem residir na mesma AWS região. 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 OSS recurso Global Datastore for Redis, 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 o modo de cluster desativado

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

### Endpoints de vários nós para Valkey ou Redis com o modo de cluster desativado OSS

Um cluster Valkey ou Redis OSS de vários nós com o modo de cluster desativado 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 (noAPI//, CLI eles são chamados de pontos finais de leitura).

### Endpoints Valkey ou Redis OSS (modo de cluster ativado)

Um OSS cluster Valkey ou Redis com o modo de cluster ativado 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 [Encontrando pontos de extremidade de conexão em ElastiCache](#).

## ElastiCache 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 [Endpoints do](#).

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

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

## ElastiCache segurança

Para aumentar a 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 (AmazonVPC). 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 aos nós, ElastiCache oferece suporte TLS e criptografia local para nós que executam versões especificadas do. ElastiCache Para obter mais informações, consulte as informações a seguir.

- [Segurança de dados na Amazon ElastiCache](#)
- [Autenticação com o comando Valkey e Redis OSS AUTH](#)

## ElastiCache grupos de sub-redes

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

Se você criar um cluster em uma AmazonVPC, deverá especificar um grupo de sub-redes de cache. 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 VPC ambiente Amazon, consulte o seguinte:

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

## ElastiCache cópias de segurança

Um backup é uma point-in-time cópia de um OSS cluster 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 do Redis em OSS execução no seu cluster, o processo de backup requer quantidades diferentes de memória reservada para ser bem-sucedido. Para obter mais informações, consulte as informações a seguir.

- [Snapshots e restauração](#)
- [Como a sincronização e o backup são implementados](#)
- [Impacto sobre o desempenho dos backups de clusters autoprojeto](#)
- [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#)

## ElastiCache eventos

Quando eventos importantes acontecem em um cluster de cache, ElastiCache envia uma notificação para um SNS tópico específico da Amazon. 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 [SNS Monitoramento de ElastiCache eventos pela Amazon](#).

# 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 no ElastiCache API e). AWS CLI Para preservar a compatibilidade com as versões anteriores, estendemos 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 no API e CLI pode ser diferente da terminologia usada no console. A lista a seguir identifica termos que podem ser diferentes entre o API e CLI e o console.

## Cluster ou nó de cache vs. nó

Há uma one-to-one relação entre um nó e um cluster de cache 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.

Os ElastiCache API e AWS CLI continuam a usar os termos como usavam no passado.

## Cluster versus grupo de replicação Valkey ou Redis OSS

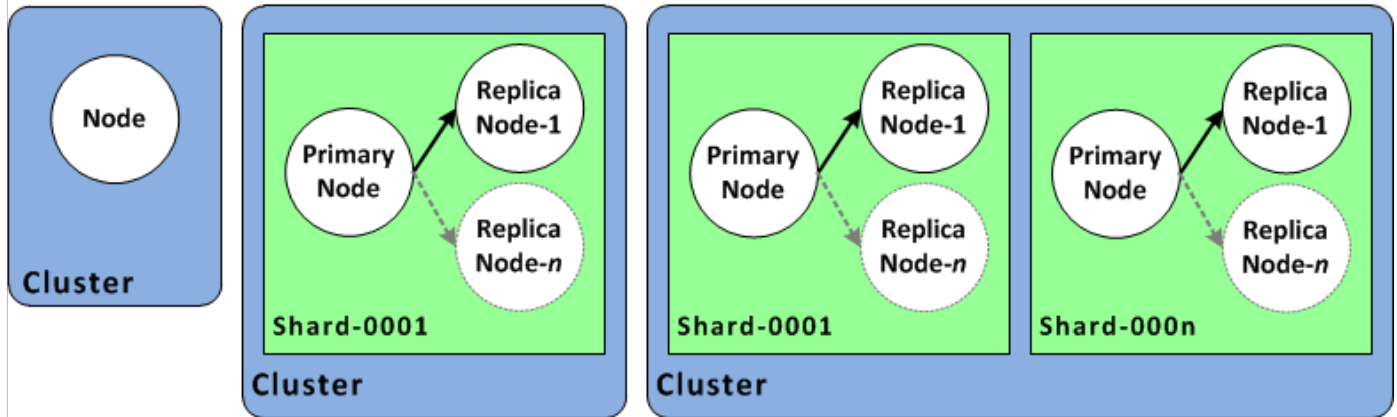
O console agora usa o termo cluster para todos os clusters ElastiCache (RedisOSS). O console usa o termo Cluster em todas essas circunstâncias:

- Quando o cluster é um cluster Valkey ou Redis OSS de nó único.
- Quando o cluster é um cluster Valkey ou Redis OSS (modo de cluster desativado) que oferece suporte à replicação em um único fragmento (no API e CLI, chamado de grupo de nós).
- Quando o cluster é um cluster Valkey ou Redis OSS (modo de cluster ativado) que oferece suporte à replicação em 1 a 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 OSS replicação Valkey ou Redis, consulte. [Alta disponibilidade com o uso de grupos de replicação](#)

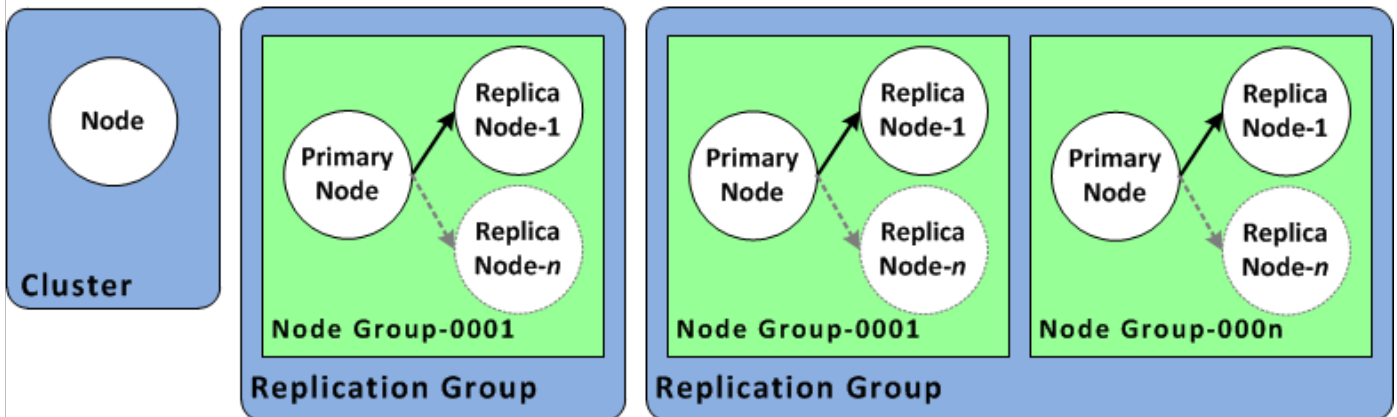
O diagrama a seguir ilustra as várias topologias dos clusters ElastiCache (RedisOSS) do ponto de vista do console.

### ElastiCache (Redis OSS): Console View



As AWS CLI operações ElastiCache API e ainda distinguem clusters de nó único ElastiCache (RedisOSS) de grupos de replicação Valkey ou OSS Redis de vários nós. O diagrama a seguir ilustra as várias topologias ElastiCache (RedisOSS) da perspectiva e. ElastiCache API AWS CLI

### ElastiCache (Redis OSS): API/CLI View



### Grupo de OSS replicação Valkey ou Redis versus armazenamento de dados global

Um armazenamento de dados global é uma coleção de um ou mais clusters que se replicam entre si em várias regiões, enquanto um grupo de OSS replicação Valkey ou Redis replica dados em um cluster habilitado para o modo de 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 armazenamentos de dados globais, consulte [Replicação entre AWS regiões usando datastores globais](#).

## Tutorial: Como criar seu próprio cluster

Veja como criar seu próprio cluster para Valkey e RedisOSS.

### Tópicos

- [Projetando seu próprio ElastiCache cluster \(Valkey\)](#)
- [Projetando seu próprio ElastiCache cluster \(RedisOSS\)](#)

## Projetando seu próprio ElastiCache cluster (Valkey)

A seguir estão as ações únicas que você deve realizar para começar a projetar seu cluster ElastiCache (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 para seus clusters de cache em um VPC. Ao iniciar um cluster de cache em um VPC, você precisa 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 [Configurando ElastiCache](#).

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 VPCID, escolha a Amazon VPC que você criou.
  - d. Nas listas Zona de disponibilidade e ID da sub-rede, escolha a Zona de disponibilidade ou [Usando zonas locais com ElastiCache](#) a ID da sua sub-rede privada e, em seguida, 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 [↗](#)

ⓘ 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 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 [Preparando um cluster em 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 lançado em uma nuvem privada virtual (VPC) com base no VPC serviço da Amazon.

Criação de um cluster Valkey (modo de cluster desativado) (console)


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

1. Faça login no AWS Management Console 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 Criar VPC e siga as etapas descritas em [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 Redis cache. 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 \(RedisOSS\)](#) com o 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 Valkey e Redis OSS](#) e [Criação de um grupo de ElastiCache parâmetros](#).

 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 [Hierarquização de dados em 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 IPv6 endereços atribuídos a eles para operar no modo de pilha dupla e uma IPv6 sub-rede somente para operar como somente. IPv4 IPv6

Ao criar um novo grupo de sub-redes, insira o VPCID ao qual ele pertence.

Para obter mais informações, consulte:

- [Escolhendo um tipo de rede em ElastiCache.](#)
- [Crie uma sub-rede no seu. VPC](#)

Se você for [Usando zonas locais com 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 [Escolhendo regiões e zonas de disponibilidade para 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 AWS KMS chave gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando chaves gerenciadas pelo cliente de 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 e para a versão 6.0 e superior OSS do mecanismo Redis, se você ativar a criptografia em trânsito, será solicitado que especifique 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 [Gerenciando grupos de usuários com o console e CLI](#).
  - AUTHUsuário padrão — Um mecanismo de autenticação para o OSS servidor Redis. Para obter mais informações, consulte [AUTH](#).
- AUTH— Um mecanismo de autenticação para o OSS servidor Redis. Para obter mais informações, consulte [AUTH](#).

 Note

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

- 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. Você pode usar o grupo de segurança padrão para o seu VPC ou criar um novo.

Para obter mais informações sobre grupos de segurança, consulte [Grupos de segurança para você VPC](#) no Guia VPC do usuário da Amazon.
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 OSS backup e restauração do Redis, consulte [Snapshots e restauração](#).

- (Opcional) Especifique uma janela de manutenção. A janela de manutenção é o horário, geralmente de uma hora de duração, a cada semana em que ElastiCache agenda a manutenção do sistema para seu cluster. Você pode ElastiCache escolher o dia e a hora da janela de manutenção (sem preferência), ou você mesmo pode escolher o dia, a hora e a duração (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 tempos são UCT tempos.

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

- (Opcional) Para Logs:
  - Em Formato de registro, escolha 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,
- 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](#).
- Escolha Next (Próximo).
- Revise todas as suas entradas e opções e faça as correções necessárias. Quando estiver pronto, escolha Create (Criar).

## On premises

- 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 o Multi-AZ
- 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 ter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4. Conecte-se ao nó do 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](#).

Criando um Valkey (modo de cluster desativado) cluster (AWS CLI)

#### Example

O CLI código a seguir cria um cluster de cache Valkey (modo de cluster desativado) sem réplicas.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine valkey \  
--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 valkey ^  
--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 Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#).

- Para usar o AWS CLI, consulte [Criando um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(AWS CLI\)](#).

## 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 (AmazonVPC), 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 mesmaVPC. A maneira mais simples de gerenciar o acesso entre EC2 instâncias e clusters da mesma forma VPC é fazer o seguinte:

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

A porta padrão para OSS clusters e grupos de replicação Valkey ou Redis 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 AmazonVPC, você controla o acesso aos seus clusters com grupos de VPC segurança da Amazon, que são diferentes dos grupos de ElastiCache segurança. Para obter mais informações sobre o uso ElastiCache em uma AmazonVPC, consulte [Amazon VPCs e ElastiCache segurança](#)

2. Crie um grupo VPC de segurança 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 VPC da tabela de roteamento. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o TCP acesso 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

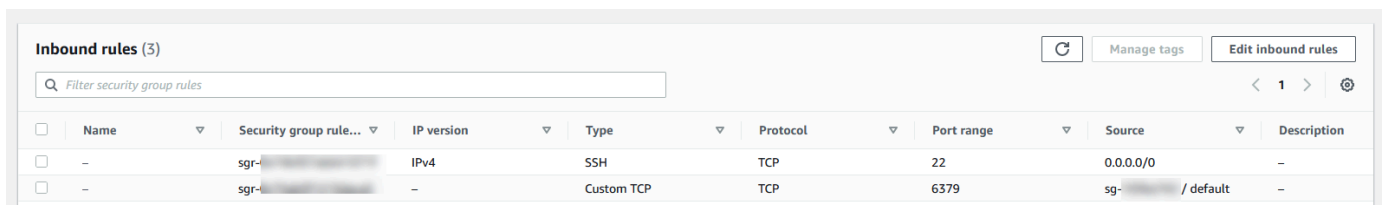
Se você estiver planejando usar [Zonas locais](#), verifique se você as habilitou. Quando você cria um grupo de sub-redes nessa zona local, ele VPC é estendido para essa zona local e você VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade. Todos os gateways e tabelas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

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

1. Faça login no AWS Management Console e abra o VPC console da Amazon em <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 TCPRegra personalizada.
  - a. Para Port Range, especifique a porta que você usou quando criou seu cluster.

A porta padrão para OSS clusters e grupos de replicação Valkey ou Redis 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.



<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-
<input type="checkbox"/>	-	sg-...	-	-	-	-	-	-

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 diferenteVPC, de uma AWS região diferente ou até mesmo de sua rede corporativa, consulte o seguinte:

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

## Etapa 4. Conecte-se 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.

### Encontrando os endpoints de um cluster Valkey (modo de cluster desativado) (console)

Se um cluster Valkey (modo de cluster desativado) tiver apenas um nó, o endpoint do nó será usado para leituras e 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 DNS nome 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 DNS registro que será resolvido para um endereço IP de um dos nós de réplica de forma redonda.

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 encontrar os endpoints de um cluster Valkey (modo de cluster desativado)

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

A tela de clusters aparecerá com uma lista que incluirá todos os caches OSS sem servidor Valkey ou Redis existentes, clusters Valkey (modo de cluster desativado) e Valkey (modo de cluster ativado). Selecione o cluster que você criou na seção [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#).

3. Para encontrar os endpoints Primário e/ou Leitor do cluster, escolha o nome do cluster (e 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 Reader para um cluster Valkey (modo de cluster desativado)

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

4. Se o cluster Valkey (modo de cluster desativado) tiver nós de réplica, você poderá encontrar os endpoints do nó de réplica do cluster escolhendo o nome do cluster e, em seguida, escolhendo a guia Nodes.



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

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	test-no-001.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-002	available	replica	6379	test-no-002.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-003	available	replica	6379	test-no-003.usw2.cache.amazonaws.com:6379

Endpoints de nós para um cluster Valkey (modo de cluster desativado)

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](#).

Um endpoint primário Valkey (modo de cluster desativado) se parece com o 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.

- [Encontrando pontos de extremidade de conexão em ElastiCache](#)

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

Conecte-se a um OSS cluster ou grupo de replicação Valkey ou Redis (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 para conectar clusters habilitados para criptografia/autenticação.

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 o `valkey-cli` com outras distribuições Linux, consulte a documentação do seu sistema operacional específico.

#### Note

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

Conexão a um cluster não criptografado com modo cluster desabilitado

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

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

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

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

2. Agora você pode executar os comandos Valkey ou RedisOSS.


```
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 e substituir *configuration-endpoint* e *port number* com o endpoint do seu cluster e o número da porta. (A porta padrão para Valkey ou Redis OSS é 6379.)

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

 Note

No comando anterior, a opção `-c` ativa o modo de cluster seguindo os redirecionamentos [-ASK](#) e [-MOVED](#).

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

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

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

```
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"
```

## Conexão a um cluster com criptografia/autenticação habilitada

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

### Note

Você pode usar a opção `--tls` com valkey-cli para se conectar aos clusters criptografados ativados e desativados no modo de cluster. Se um cluster tiver um conjunto de AUTH tokens, você poderá usar a opção `-a` de fornecer uma AUTH senha.

Nos exemplos a seguir, certifique-se de substituir *cluster-endpoint* e *port number* com o endpoint do seu cluster e o número da porta. (A porta padrão para 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 pode executar os OSS comandos Valkey ou Redis, conforme mostrado nos exemplos anteriores para clusters não criptografados.

#### alternativa valkey-cli

Se o cluster não estiver habilitado no modo cluster e você precisar fazer uma conexão com o cluster para um breve teste, mas sem passar pela compilação 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 porta. (A porta padrão para Valkey ou Redis OSS é 6379.)

O exemplo a seguir conecta a um cluster habilitado para criptografia e/ou autenticação com modo cluster desabilitado:

```
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, substitua *your-password* com a 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
```

#### Conecte-se a um OSS cluster ou grupo de replicação Valkey ou Redis (Windows)

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

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

Para se conectar a um OSS cluster Valkey ou Redis que não esteja habilitado para criptografia usando `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, vá para o diretório Valkey e execute o comandoc :

```
\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 RedisOSS.

Agora você está conectado ao cluster e pode executar OSS comandos Valkey ou Redis 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](#)
- [ElastiCache APIReferência 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.

- [ElastiCache melhores práticas e estratégias de armazenamento em cache](#)

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

## Projetando seu próprio ElastiCache cluster (RedisOSS)

A seguir estão as ações únicas que você deve realizar para criar seu próprio cluster ElastiCache (RedisOSS).

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

### 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 para seus clusters de cache em um VPC. Ao iniciar um cluster de cache em um VPC, você precisa 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 VPCID, 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**

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 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 [Preparando um cluster em 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 lançado em uma nuvem privada virtual (VPC) com base no VPC serviço da Amazon.

Criação de um cluster Redis OSS (modo de cluster desativado) (console)


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

1. Faça login no AWS Management Console 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 Criar VPC e siga as etapas descritas em [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 OSScache do Redis e, em seguida, escolheremos Criar cache do 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 \(RedisOSS\)](#) com o 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 Valkey e Redis OSS](#) e [Criação de um grupo de ElastiCache parâmetros](#).

 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 [Hierarquização de dados em 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 IPv6 endereços atribuídos a eles

para operar no modo de pilha dupla e uma IPv6 sub-rede somente para operar como somente. IPv4 IPv6

Ao criar um novo grupo de sub-redes, insira o VPCID ao qual ele pertence.

Para obter mais informações, consulte:

- [Escolhendo um tipo de rede em ElastiCache.](#)
- [Crie uma sub-rede no seu. VPC](#)

Se você for [Usando zonas locais com 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 [Escolhendo regiões e zonas de disponibilidade para ElastiCache.](#)

5. Escolha Next (Próximo)
6. Em Configurações avançadas 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 [AWS KMS](#) chave gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando chaves gerenciadas pelo cliente de 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 e superior OSS do mecanismo Redis, 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 [Gerenciando grupos de usuários com o console e CLI](#).
  - AUTHUsuário padrão — Um mecanismo de autenticação para servidores Valkey e RedisOSS. Para obter mais informações, consulte [AUTH](#).
- AUTH— Um mecanismo de autenticação para o OSS servidor Redis. Para obter mais informações, consulte [AUTH](#).

**Note**

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

- 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. Você pode usar o grupo de segurança padrão para o seu VPC ou criar um novo.

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

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 janela de manutenção é o horário, geralmente de uma hora de duração, a cada semana em que ElastiCache agenda a manutenção do sistema para seu cluster. Você pode ElastiCache escolher o dia e a hora da janela de manutenção (sem preferência), ou você mesmo pode escolher o dia, a hora e a duração (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 tempos são UCT tempos.

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

9. (Opcional) Para Logs:
  - Em Formato de registro, escolha 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 o 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 ter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4. Conecte-se ao nó do 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](#).

## Criando um cluster Redis OSS (modo de cluster desativado) (AWS CLI)

### Example

O CLI código a seguir cria um cluster de cache Redis OSS (modo de cluster desativado) 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 Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#).
- Para usar o AWS CLI, consulte [Criando um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(AWS CLI\)](#).

## 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 (AmazonVPC), 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 lançado seu cluster no EC2 - VPC ou no EC2 -Classic.

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

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

A porta padrão para OSS clusters e grupos de replicação do Redis é. 6379

### Important

Os grupos ElastiCache de segurança da Amazon são aplicáveis somente aos clusters que não estão sendo executados em um ambiente da 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 AmazonVPC, você controla o acesso aos seus clusters com grupos de VPC segurança da Amazon, que são diferentes dos grupos de ElastiCache segurança. Para obter mais informações sobre o uso ElastiCache em uma AmazonVPC, consulte [Amazon VPCs e ElastiCache segurança](#)

2. Crie um grupo VPC de segurança 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 VPC da tabela de roteamento. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o TCP acesso 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

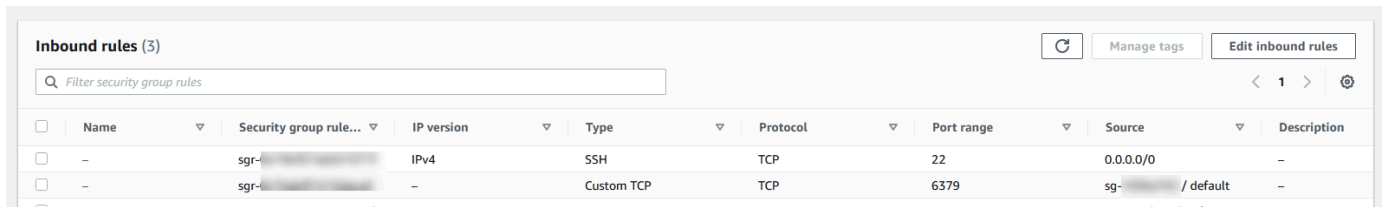
Se você planeja usar [Usando zonas locais com ElastiCache](#), certifique-se de tê-los habilitado. Quando você cria um grupo de sub-redes nessa zona local, ele VPC é estendido para essa zona local e você VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade. Todos os gateways e tabelas de rotas relevantes serão ajustados automaticamente.

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

1. Faça login no AWS Management Console e abra o VPC console da Amazon em <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 TCPRegra personalizada.
  - a. Para Port Range, especifique a porta que você usou quando criou seu cluster.

A porta padrão para OSS clusters e grupos de replicação do Redis 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.



Inbound rules (3)

Filter security group rules

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-...	-	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 diferente VPC, de uma AWS região diferente ou até mesmo de sua rede corporativa, consulte o seguinte:

- [Padrões de acesso para acessar um ElastiCache cache em uma 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.

Encontrando os endpoints de um cluster Valkey ou Redis OSS (modo de cluster desativado) (console)

Se um cluster Redis OSS (modo de cluster desativado) tiver somente um nó, o endpoint do nó será usado para leituras e 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 DNS nome 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 (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 as várias réplicas de leitura do seu cluster ElastiCache (RedisOSS) 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 DNS registro que será resolvido para um endereço IP de um dos nós de réplica de forma redonda.

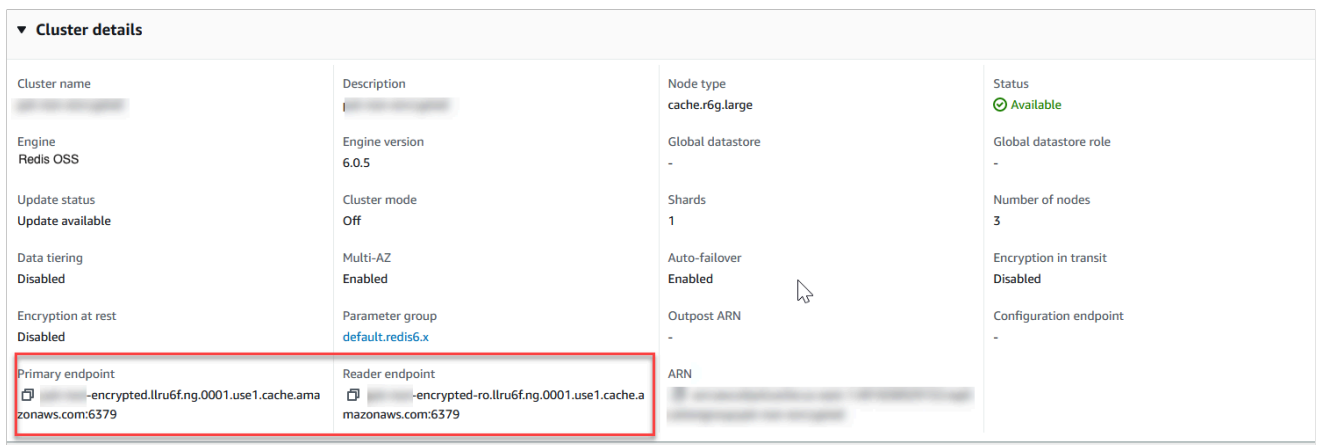
Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó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

Para encontrar os endpoints de um cluster Redis OSS (modo de cluster desativado)

1. Faça login no AWS Management Console 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 incluirá todos os caches OSS sem servidor Valkey ou Redis existentes, clusters Redis (modo de cluster desativado) e clusters Redis OSS (modo de cluster ativado). OSS Selecione o cluster que você criou na seção [Criação de um cluster Redis OSS \(modo de cluster desativado\) \(console\)](#).

3. Para encontrar os endpoints Primário e/ou Leitor do cluster, escolha o nome do cluster (e não o botão de opção).



▼ Cluster details			
Cluster name	Description	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 [icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Endpoints primários e Reader para um cluster Redis OSS (modo de cluster desativado)

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

4. Se o cluster Redis OSS (modo de cluster desativado) tiver nós de réplica, você poderá encontrar os endpoints do nó de réplica do cluster escolhendo o nome do cluster e, em seguida, escolhendo a guia Nodes.

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

<input type="checkbox"/>	Node Name	▲	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001		available	primary	6379	redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-002		available	replica	6379	redis-01.7abc2d.0002.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-003		available	replica	6379	redis-01.7abc2d.0003.usw2.cache.amazonaws.com:6379

Endpoints de nós para um cluster Redis OSS (modo de cluster desativado)

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](#).

Um endpoint primário do Redis OSS (modo de cluster desativado) se parece com o 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.

- [Encontrando pontos de extremidade de conexão em ElastiCache](#)

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

Conecte-se a um OSS cluster ou grupo de replicação Valkey ou Redis (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 para conectar clusters habilitados para criptografia/autenticação.

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 o `valkey-cli` com outras distribuições Linux, consulte a documentação do seu sistema operacional específico.

#### Note

Esse processo abrange o teste de uma conexão usando o utilitário `valkey-cli` somente para uso não planejado. Para obter uma lista de clientes compatíveis, consulte a [documentação do Valkey](#). Para obter exemplos de uso do AWS SDKs com ElastiCache, consulte [Tutoriais: Introdução ao Python e ElastiCache](#).

Conexão a um cluster não criptografado com modo cluster desabilitado

1. Execute o comando a seguir para se conectar ao cluster e substituir *primary-endpoint* e *port number* com o endpoint do seu cluster e o número da porta. (A porta padrão para 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 os comandos Valkey e RedisOSS.




```
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 e substituir *configuration-endpoint* e *port number* com o endpoint do seu cluster e o número da porta. (A porta padrão para Valkey e Redis OSS é 6379.)

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

 Note

No comando anterior, a opção `-c` ativa o modo de cluster seguindo os redirecionamentos [-ASK](#) e [-MOVED](#).

O resultado em um prompt de comando é semelhante ao seguinte:

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

2. Agora você pode executar os comandos Valkey e RedisOSS. Observe que o redirecionamento ocorre porque você o habilitou usando a opção `-c`. Se o redirecionamento não estiver ativado, o comando retornará o MOVED erro. Para obter mais informações sobre o MOVED erro, consulte a [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"
```

## Conexão a um cluster com criptografia/autenticação habilitada

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

### Note

Você pode usar a opção `--tls` com valkey-cli para se conectar aos clusters criptografados ativados e desativados no modo de cluster. Se um cluster tiver um conjunto de AUTH tokens, você poderá usar a opção `-a` de fornecer uma AUTH senha.

Nos exemplos a seguir, certifique-se de substituir *cluster-endpoint* e *port number* com o endpoint do seu cluster e o número da porta. (A porta padrão para 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 pode executar os OSS comandos Valkey ou Redis, conforme mostrado nos exemplos anteriores para clusters não criptografados.

Uma alternativa ao valkey-cli ou ao Redis-cli

Se o cluster não estiver habilitado no modo cluster e você precisar fazer uma conexão com o cluster para um breve teste, mas sem passar pela compilação 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 porta. (A porta padrão para Redis OSS é 6379.)

O exemplo a seguir conecta a um cluster habilitado para criptografia e/ou autenticação com modo cluster desabilitado:

```
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, substitua *your-password* com a 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
```

Conecte-se a um OSS cluster ou grupo de replicação Valkey ou Redis (Windows)

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

No exemplo a seguir, você usa o utilitário `valkey-cli` para se conectar a um cluster que não tem criptografia habilitada e está executando 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 OSS cluster Valkey ou Redis que não esteja habilitado para criptografia usando `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 OSS cliente Redis 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, vá para o diretório Valkey e execute o comando:

```
\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379.
```

Por exemplo:

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

3. Execute os comandos Valkey ou RedisOSS.

Agora você está conectado ao cluster e pode executar OSS comandos Valkey ou Redis 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](#).
- `CreateSnapshot` é necessária permissão para criar um instantâneo final. Sem essa permissão, a API chamada falhará com uma `Access Denied` exceção.

## Usando o AWS Management Console

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 AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache motor, escolha Valkey ou RedisOSS.

Uma lista de todos os caches em execução nesse 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 de cache `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 `delete-cache-cluster` CLI ação exclui somente um cluster de cache. Para excluir vários clusters de cache, chame `delete-cache-cluster` para cada cluster de cache que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster de cache 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 ElastiCache tutoriais e vídeos

Os tutoriais a seguir abordam tarefas de interesse para o usuário da Amazon ElastiCache .

- [ElastiCache Vídeos](#)
- [Tutorial: Configurando uma função Lambda para acessar a Amazon em uma ElastiCache Amazon VPC](#)

## ElastiCache Vídeos

Aqui estão os vídeos para ajudar você a aprender os ElastiCache conceitos básicos e avançados da Amazon. Para obter informações sobre AWS treinamento, consulte [AWS Treinamento e certificação](#).

### Tópicos

- [Vídeos de introdução](#)
- [Vídeos avançados](#)

### Vídeos de introdução

Os vídeos a seguir apresentam a Amazon ElastiCache.

### Tópicos

- [AWS re:Invent 2020: O que há de novo na Amazon ElastiCache](#)
- [AWS re:Invent 2019: O que há de novo na Amazon ElastiCache](#)
- [AWS re:Invent 2017: O que há de novo na Amazon ElastiCache](#)
- [DAT204—Criação de aplicativos escaláveis AWS sem SQL serviços \(re:Invent 2015\)](#)
- [DAT207 — Acelerando o desempenho de aplicativos com a Amazon ElastiCache \(re:Invent 2013\)AWS](#)

AWS re:Invent 2020: O que há de novo na Amazon ElastiCache

[AWS re:Invent 2020: O que há de novo na Amazon ElastiCache](#)

AWS re:Invent 2019: O que há de novo na Amazon ElastiCache

[AWS re:Invent 2019: O que há de novo na Amazon ElastiCache](#)

AWS re:Invent 2017: O que há de novo na Amazon ElastiCache

[AWS re:Invent 2017: O que há de novo na Amazon ElastiCache](#)

DAT204—Criação de aplicativos escaláveis AWS sem SQL serviços (re:Invent 2015)

Nesta sessão, discutimos os benefícios da ausência de SQL bancos de dados e fazemos um tour pelos principais SQL serviços sem bancos de dados oferecidos pelo AWS— Amazon DynamoDB



e Amazon. ElastiCache Em seguida, ouvimos dois clientes líderes, Expedia e Mapbox, sobre seus casos de uso e desafios arquitetônicos, e como eles os abordaram usando AWS nenhum SQL serviço, incluindo padrões de design e melhores práticas. Você deve sair desta sessão com uma melhor compreensão do No SQL e de seus poderosos recursos, pronto para enfrentar os desafios do banco de dados com confiança.

### [DAT204—Criação de aplicativos escaláveis AWS sem SQL serviços \(re:Invent 2015\)](#)

DAT207 — Acelerando o desempenho de aplicativos com a Amazon ElastiCache (re:Invent 2013)AWS

Neste vídeo, saiba como você pode usar ElastiCache a Amazon para implantar facilmente um sistema de cache na memória para acelerar o desempenho do seu aplicativo. Mostramos como usar a Amazon para melhorar ElastiCache a latência do aplicativo e reduzir a carga nos 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 ativação do armazenamento em cache e discutimos os recursos fornecidos pela Amazon ElastiCache.

### [DAT207 - Acelerando o desempenho de aplicativos com a Amazon ElastiCache \(re:Invent 2013\)](#)

## Vídeos avançados

Os vídeos a seguir ElastiCache abordam tópicos mais avançados da Amazon.

### Tópicos

- [Crie para o sucesso com as ElastiCache melhores práticas da Amazon \(re:Invent 2020\)](#)
- [Potencialize seus aplicativos em tempo real com a Amazon ElastiCache \(re:Invent 2019\)](#)
- [Melhores práticas: migrar OSS clusters Redis da Amazon EC2 para ElastiCache \(re:Invent 2019\)](#)
- [Escalando uma plataforma Fantasy Sports com a Amazon ElastiCache e a Amazon STP11 Aurora \(re:Invent 2018\)](#)
- [Redis confiável e escalável OSS na nuvem com a Amazon ElastiCache \(re:Invent 2018\)](#)
- [ElastiCache Deep Dive: padrões de design para armazenamentos de dados na memória \(re:Invent 2018\)](#)
- [DAT305 — Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)
- [DAT306 — Amazon ElastiCache Deep Dive \(re:Invent 2016\)](#)

- [DAT317—Como IFTTT usa ElastiCache \(RedisOSS\) para prever eventos \(re:Invent 2016\)](#)
- [DAT407 — Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)
- [SDD402 — Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)
- [DAT307 — Mergulhe profundamente nos padrões de ElastiCache arquitetura e design da Amazon \(re:Invent 2013\)](#)

[Crie para o sucesso com as ElastiCache melhores práticas da Amazon \(re:Invent 2020\)](#)

Com o crescimento explosivo de aplicativos essenciais para os negócios em tempo real desenvolvidos no Redis, a disponibilidadeOSS, a escalabilidade e a segurança se tornaram as principais considerações. Conheça as melhores práticas para configurar a Amazon ElastiCache para o sucesso com escalabilidade on-line, alta disponibilidade em implantações Multi-AZ e configurações de segurança.

[Crie para o sucesso com as ElastiCache melhores práticas da Amazon \(re:Invent 2020\)](#)

[Potencialize seus aplicativos em tempo real com a 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 você pode aumentar o desempenho de aplicativos em tempo real usando a Amazon totalmente gerenciada e em memória ElastiCache para obter desempenho extremo, alta escalabilidade, disponibilidade e segurança.

[Potencialize seus aplicativos em tempo real com a Amazon ElastiCache \(re:Invent 2019:\)](#)

[Melhores práticas: migrar OSS clusters Redis da Amazon EC2 para ElastiCache \(re:Invent 2019\)](#)

Gerenciar OSS clusters do Redis sozinho pode ser difícil. Você precisa provisionar hardware, corrigir software, fazer backup de dados e monitorar cargas de trabalho constantemente. Com o recurso de migração on-line recém-lançado para a Amazon ElastiCache, agora você pode mover facilmente seus dados do Redis auto-hospedado na Amazon EC2 para a OSS Amazon totalmente gerenciada ElastiCache, com o modo de cluster desativado. Nesta sessão, você aprenderá sobre a nova ferramenta de migração on-line, assistirá a uma demonstração e, o mais importante, aprenderá as melhores práticas para uma migração tranquila para a Amazon. ElastiCache

## [Melhores práticas: migrar OSS clusters Redis da Amazon EC2 para ElastiCache \(re:Invent 2019\)](#)

Escalando uma plataforma Fantasy Sports com a Amazon ElastiCache e a Amazon STP11 Aurora (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, CTO Dream11 Amit Sharma explica como a empresa usa o Amazon Aurora e o Amazon ElastiCache para lidar com o tráfego flash, 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: escalando uma plataforma de esportes de fantasia com Amazon e Amazon ElastiCache Aurora () STP11

## [Escalando uma plataforma Fantasy Sports com a Amazon ElastiCache e a Amazon STP11 Aurora \(re:Invent 2018\)](#)

Redis confiável e escalável OSS na nuvem com a Amazon ElastiCache (re:Invent 2018)

Esta sessão aborda os recursos e aprimoramentos em nosso OSS serviço compatível com o Redis, o Amazon ElastiCache (Redis). OSS Abordamos os principais recursos, como o Redis OSS 5, melhorias de escalabilidade e desempenho, segurança e conformidade e muito mais. Também discutimos recursos futuros e estudos de caso de clientes.

## [Redis confiável e escalável OSS na nuvem com a Amazon ElastiCache \(re:Invent 2018\)](#)

ElastiCache Deep Dive: padrões de design para armazenamentos de dados na memória (re:Invent 2018)

Nesta sessão, damos uma olhada nos bastidores para aprender sobre o design e a arquitetura da Amazon ElastiCache. Veja padrões de design comuns com nossas ofertas de Redis OSS e Memcached e como os clientes os usam para processamento de dados na memória para reduzir a latência e melhorar a produtividade do aplicativo. Analisamos as ElastiCache melhores práticas, padrões de design e antipadrões.

## [ElastiCache Deep Dive: padrões de design para armazenamentos de dados na memória \(re:Invent 2018\)](#)

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

Veja os bastidores para aprender sobre o design e a arquitetura ElastiCache da Amazon. Veja padrões de design comuns com nossas OSS ofertas de Memcached e Redis e como os clientes os usaram em operações na memória para reduzir a latência e melhorar a taxa de transferência do aplicativo. Durante este vídeo, analisamos as ElastiCache melhores práticas, padrões de design e antipadrões.

O vídeo apresenta o seguinte:

- ElastiCache (RedisOSS) refragmentação online
- ElastiCache segurança e criptografia
- ElastiCache (RedisOSS) versão 3.2.10

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

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

Veja os bastidores para aprender sobre o design e a arquitetura ElastiCache da Amazon. Veja padrões de design comuns com nossas OSS ofertas de Memcached e Redis e como os clientes os usaram em operações na memória para reduzir a latência e melhorar a taxa de transferência do aplicativo. Durante esta sessão, analisamos as ElastiCache melhores práticas, padrões de design e antipadrões.

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

## DAT317—Como IFTTT usa ElastiCache (RedisOSS) para prever eventos (re:Invent 2016)

IFTTT é um serviço gratuito que capacita as pessoas a fazerem mais com os serviços que adoram, desde automatizar tarefas simples até transformar a forma como alguém interage e controla sua casa. IFTTT usa ElastiCache (RedisOSS) para armazenar o histórico de execução de transações e programar previsões, bem como índices para documentos de log no Amazon S3. Assista a esta sessão para saber como o poder de criação de scripts de Lua e os tipos de dados do Redis OSS permitiram que as pessoas realizassem algo que não conseguiriam em outro lugar.

### [DAT317—Como IFTTT usa ElastiCache \(RedisOSS\) para prever eventos \(re:Invent 2016\)](#)

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

Dê uma olhada nos bastidores para aprender sobre o design e a arquitetura ElastiCache da Amazon. Veja os padrões de design comuns de nossas OSS ofertas de Memcached e Redis e como os

clientes os usaram para operações na memória e obtiveram maior latência e taxa de transferência para aplicativos. Durante esta sessão, analisamos as melhores práticas, padrões de design e antipadrões relacionados à Amazon ElastiCache.

### [DAT407 — Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)

### [SDD402 — Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)

Neste vídeo, examinamos casos de uso comuns de cache, os OSS mecanismos Memcached e Redis, padrões que ajudam você a determinar qual mecanismo é melhor para suas necessidades, hashing consistente e muito mais como meio de criar aplicativos rápidos e escaláveis. Frank Wiebe, cientista principal da Adobe, detalha como a Adobe usa ElastiCache a Amazon para melhorar a experiência do cliente e expandir seus negócios.

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

DAT307 — Mergulhe profundamente nos padrões de ElastiCache arquitetura e design da Amazon (re:Invent 2013)

Neste vídeo, examinamos o armazenamento em cache, as estratégias de cache, a expansão e o monitoramento. Também comparamos os mecanismos Memcached e RedisOSS. Durante esta sessão, também analisamos as melhores práticas e padrões de design relacionados à Amazon ElastiCache.

### [DAT307 - Mergulhe profundamente nos padrões de ElastiCache arquitetura e design da Amazon \(AWS re:Invent 2013\).](#)

## Gerenciando nós em ElastiCache

Um nó é o menor componente básico de uma ElastiCache implantação da Amazon. É um pedaço de tamanho fixo, seguro e conectado à rede. RAM Cada nó executa o mecanismo que foi escolhido quando o cluster foi criado ou modificado pela última vez. Cada nó tem seu próprio nome e porta do Domain Name Service (DNS). Vários tipos de ElastiCache nós são suportados, cada um com quantidades variáveis de memória associada e poder computacional.

Consulte [Escolha do tamanho do nó](#) para uma discussão mais detalhada sobre o tamanho do nó que deve ser usado.

De um modo geral, devido ao suporte para fragmentação, as implantações do Valkey ou Redis OSS (modo de cluster ativado) têm vários nós menores. Por outro lado, as implantações do Valkey ou do

Redis OSS (modo de cluster desativado) têm menos nós maiores em um cluster. Consulte [Escolha do tamanho do nó](#) para uma discussão mais detalhada sobre o tamanho do nó que deve ser usado.

## Tópicos

- [Visualizando o status ElastiCache do nó](#)
- [OSSNódulos e fragmentos Valkey ou Redis](#)
- [Conexão a nós](#)
- [Tipos de nó compatíveis](#)
- [Reinicializando nós](#)
- [Substituindo nós \(Valkey e RedisOSS\)](#)
- [Substituindo nós \(Memcached\)](#)
- [Nós reservados](#)
- [Migração de nós da geração anterior](#)

Algumas operações importantes que envolvem nós são as seguintes:

- [Adicionando nós a um ElastiCache cluster](#)
- [Removendo nós de um ElastiCache cluster](#)
- [Dimensionamento ElastiCache](#)
- [Encontrando pontos de extremidade de conexão em ElastiCache](#)
- [Identifique automaticamente os nós em 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 API operação.

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 nóculo 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	<p>Um estado de atualização indica que uma ou mais das seguintes afirmações são verdadeiras para o ElastiCache nó da Amazon:</p> <ul style="list-style-type: none"><li>• O ElastiCache nó está sendo corrigido como parte da atualização do serviço. Para obter mais informações sobre as atualizações do serviço, consulte a <a href="#">página de ajuda do Amazon ElastiCache Managed Maintenance and Service Updates</a>.</li><li>• Os grupos VPC de segurança estão sendo atualizados para o ElastiCache cluster.</li></ul>

Tipo	Faturado	Descrição
		<ul style="list-style-type: none"><li>• O ElastiCache cluster está sendo <a href="#">ampliado ou reduzido</a>.</li><li>• As <a href="#">configurações de entrega de registros</a> estão sendo modificadas para o ElastiCache cluster.</li><li>• Uma operação de exclusão do ElastiCache nó está pendente.</li><li>• A OSS senha ElastiCache com Valkey ou Redis está sendo atualizada/rotacionada usando <a href="#">AWS Secrets Manager</a></li></ul>
rebooting cache 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
incompatible_parameters	Não faturado	<p>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 <a href="#">eventos</a> do ElastiCache nó.</p>
incompatible_network	Não faturado	<p>Um estado de rede incompatível indica que uma ou mais das seguintes afirmações são verdadeiras para o nó da Amazon: ElastiCache</p> <ul style="list-style-type: none"><li>• Não há endereços IP disponíveis na sub-rede em que o ElastiCache nó foi lançado.</li><li>• A sub-rede mencionada no grupo de ElastiCache sub-redes não existe mais na Amazon Virtual Private Cloud (AmazonVPC).</li></ul>

Tipo	Faturado	Descrição
restore_failed	Não faturado	<p>Um estado de falha na restauração indica que uma das seguintes situações se aplica ao nó da Amazon ElastiCache</p> <ul style="list-style-type: none"><li>• As substituições de nó com falha por causa da <a href="#">capacidade da instância insuficiente</a>. 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 <a href="#">Migração de nós da geração anterior</a>.</li><li>• Falha na restauração do RDB instantâneo especificado.</li><li>• A AWS conta do ElastiCache cluster foi suspensa.</li><li>• O nó falhou e não pôde ser recuperado.</li></ul>

Tipo	Faturado	Descrição
snapshotting	Faturado	ElastiCache está criando um instantâneo do nó Valkey ou RedisOSS.

## Visualizando o status do ElastiCache nó com o console

Para ver o status de um ElastiCache Node com o console:

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

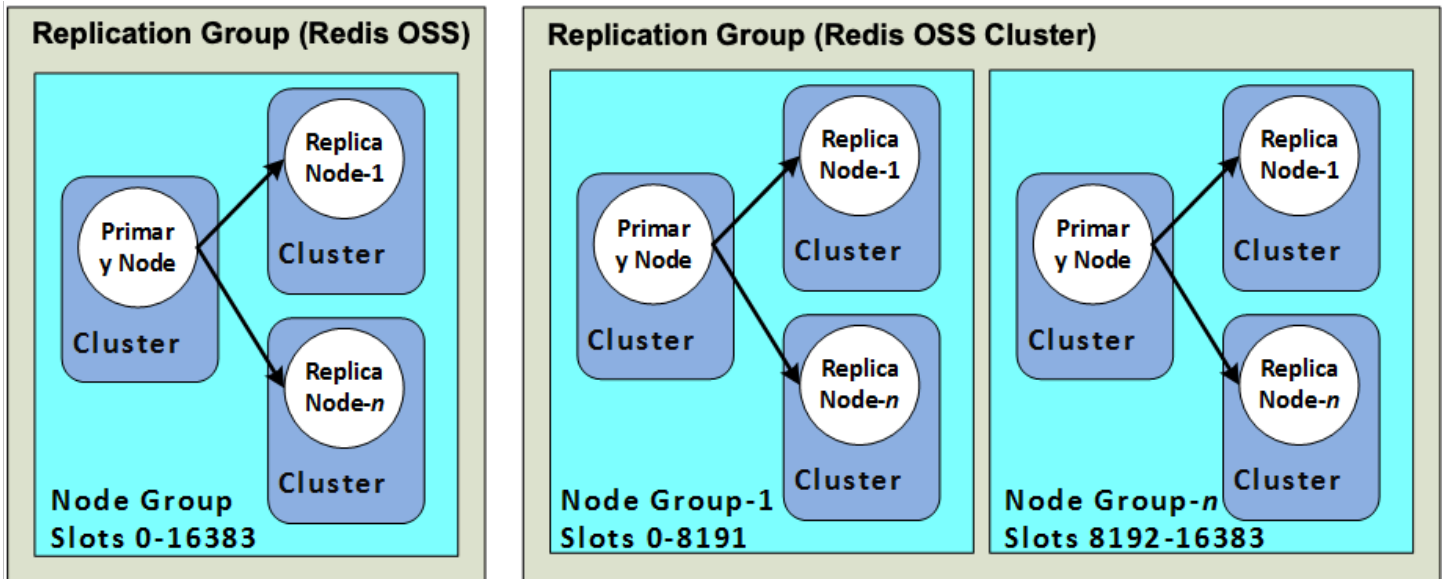
Para ver o status do ElastiCache nó usando o Amazon ElastiCache API, chame `DescribeCacheClusteroperation` with the `ShowCacheNodeInfo` flag para recuperar informações sobre os nós de cache individuais.

## OSSNódulos e fragmentos Valkey ou Redis

Um fragmento (em API eCLI, um grupo de nós) é um arranjo hierárquico de nós, cada um agrupado 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 Redis OSS versão 3.2 e posterior,

oferece suporte a vários fragmentos em um cluster (no API eCLI, em um grupo de replicação). Esse suporte permite particionar seus dados em um cluster Valkey ou Redis OSS (modo de cluster ativado).

O diagrama a seguir ilustra as diferenças entre um cluster Valkey ou Redis OSS (modo de cluster desativado) e um cluster Valkey ou Redis OSS (modo de cluster ativado).



Os clusters Valkey ou Redis OSS (modo de cluster ativado) oferecem suporte à replicação por meio de fragmentos. A API operação [DescribeReplicationGroups](#)(CLI: [describe-replication-groups](#)) lista os grupos de nós com os nós membros, a função do nó dentro do grupo de nós e também outras informações.

Ao criar um OSS cluster Valkey ou Redis, você especifica se deseja criar um cluster com o clustering ativado. Os clusters Valkey ou Redis OSS (modo de cluster desativado) nunca têm mais de um fragmento, que pode ser escalado horizontalmente adicionando (até um total de cinco) ou excluindo nós de réplica de leitura. Para mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#), [Adicionar uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#) ou [Excluindo uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#). Os clusters Valkey ou Redis OSS (modo de cluster desativado) também podem ser escalados verticalmente alterando os tipos de nós. Para obter mais informações, consulte [Dimensionamento de nós de réplica para Valkey ou Redis OSS \(modo de cluster desativado\)](#).

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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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 Valkey ou Redis OSS (modo de cluster ativado) é criado, ele pode ser alterado (ampliado ou reduzido). Para ter mais informações, consulte [Dimensionamento ElastiCache](#) e [Substituindo nós \(Valkey e RedisOSS\)](#).

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 ter mais informações, consulte [Realização de backups manuais](#) e [Restauração de um backup para um novo cache](#).

## Conexão a nós

### Conectando-se aos nós Valkey ou Redis OSS

Antes de tentar se conectar aos OSS nós Valkey ou Redis em seu cluster, você deve ter os endpoints para os nós. Para localizar os endpoints, consulte o seguinte:

- [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
- [Encontrando endpoints para um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)
- [Localizar endpoints \(AWS CLI\)](#)
- [Encontrando endpoints \(\) ElastiCache API](#)

No exemplo a seguir, você usa o utilitário `valkey-cli` para se conectar a um cluster que está executando Valkey ou Redis. OSS

#### Note

Para obter mais informações sobre os comandos disponíveis, consulte a página [Comandos](#) na web.

Para se conectar a um OSS cluster Valkey ou Redis usando o `valkey-cli`

1. Conecte-se à sua EC2 instância da Amazon usando o utilitário de conexão de sua escolha.

#### Note

Para obter instruções sobre como se conectar a uma EC2 instância da Amazon, consulte o [Amazon EC2 Getting Started Guide](#).

2. Para criar `valkey-cli`, baixar e instalar o GNU Compiler Collection (`gcc`). No prompt de comando da sua EC2 instância, insira o comando a seguir e entre `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 EC2 instância, 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/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean      # ubuntu systems only
make
```

4. No prompt de comando da sua EC2 instância, digite o comando a seguir.

```
src/valkey-cli -c -h mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

Um prompt de OSS comando Valkey ou Redis semelhante ao seguinte é exibido.

```
redis mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

## 5. Teste a conexão executando os comandos Valkey ou RedisOSS.

Agora você está conectado ao cluster e pode executar os comandos Valkey ou RedisOSS. A seguir estão alguns exemplos de comandos com suas respostas Valkey ou RedisOSS.

```
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 que têm criptografia Secure Sockets Layer (SSL) (ativada em trânsito), consulte [ElastiCache criptografia em trânsito \(\) TLS](#)

## Conectando-se aos 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:

- [Encontrando os endpoints de um cluster \(console\) \(Memcached\)](#)
- [Localizar endpoints \(AWS CLI\)](#)
- [Encontrando endpoints \(\) ElastiCache API](#)

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 sua EC2 instância da Amazon usando o utilitário de conexão de sua escolha.

**Note**

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 telnet na sua EC2 instância da Amazon. No prompt de comando da sua EC2 instância da Amazon, digite o comando a seguir e digite y no prompt de comando.

```
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 EC2 instância da Amazon, digite o comando a seguir, substituindo o endpoint do seu nó pelo 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](#).

Para obter informações sobre o tamanho do nó que deve ser usado, consulte [Escolha do tamanho do nó](#).

### Tópicos

- [Geração atual \(Memcached\)](#)
- [Geração atual \(Valkey ou RedisOSS\)](#)
- [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 de linha de base e de expansão que utilizam o mecanismo de crédito de E/S de rede para ultrapassar a largura de banda de linha de base.

#### Note

Tipos de instância com performance de rede expansível utilizam um mecanismo de crédito de E/S de rede para ultrapassar a respectiva largura de banda de linha de base conforme o 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
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

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
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
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 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

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
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
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

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
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 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
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 RedisOSS)

Para obter mais informações sobre a geração anterior, consulte [Nós da geração anterior](#).

**Note**

Tipos de instância com performance de rede expansível utilizam um mecanismo de crédito de E/S de rede para ultrapassar a respectiva largura de banda de linha de base conforme o melhor esforço.

## Geral

Tipo de instância	Versão mínima suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLSDescarregando (Redis OSS 6.2.5+)	Múltipla largura de banda de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda de 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
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



Tipo de instância	Versão mínima suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLS Descartando (Redis OSS 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda base (Gbps)	Expansão da largura de banda (Gbps)
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
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

Tipo de instância	Versão mínima suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLSDescarregando (Redis OSS 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
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
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

Tipo de instância	Versão mínima suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLSDescarregando (Redis OSS 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda base (Gbps)	Expansão da largura de banda (Gbps)
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 suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLSDescarregando (Redis OSS 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda 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

Tipo de instância	Versão mínima suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLS Descartando (Redis OSS 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda base (Gbps)	Expansão da largura de banda (Gbps)
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
cache.r6g .8xlarge	5.0.6	S	S	S	12	N/D
cache.r6g .12xlarge	5.0.6	S	S	S	20	N/D

Tipo de instância	Versão mínima suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLSDescarregando (Redis OSS 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda base (Gbps)	Expansão da largura de banda (Gbps)
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
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 suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLS Descartando (Redis OSS 6.2.5+)	Múltipla conexão de E/S aprimorada (Redis 7.0.4+OSS)	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
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 suportada do Redis OSS	E/S aprimorada (Redis 5.0.6+OSS)	TLSDescarregando (Redis OSS 6.2.5+)	Múltipla ação de E/S aprimorada (Redis 7.0.4+OSS)	Largura de banda de linha 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
cache.c7gn.2xlarge	6.2	S	S	S	25	50
cache.c7gn.4xlarge	6.2	S	S	S	50	N/D
cache.c7gn.8xlarge	6.2	S	S	S	100	N/D
cache.c7gn.12xlarge	6.2	S	S	S	150	N/D
cache.c7gn.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 básico de CPU desempenho com a capacidade de aumentar o CPU uso a qualquer momento até que os créditos acumulados se esgotem. Um CPUcrédito fornece o desempenho de um CPU núcleo completo por um minuto.

ElastiCacheOs nós T4G, T3 e T2 da Amazon são configurados como padrão e adequados para cargas de trabalho com uma CPU utilização média consistentemente abaixo do desempenho básico da instância. Para ultrapassar a linha de base, o nó gasta créditos acumulados em seu saldo de crédito. 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 sofra uma queda acentuada no desempenho quando o saldo de crédito acumulado estiver esgotado. CPU Para obter mais informações, consulte [CPUcréditos e desempenho básico para instâncias de desempenho intermitentes no Guia EC2](#) do usuário da Amazon.

A tabela a seguir lista os tipos de nós de desempenho com capacidade de intermitência, a taxa na qual os CPU créditos são ganhos por hora. Também mostra o número máximo de CPU créditos ganhos que um nó pode acumular e o número de créditos vCPUs por nó. Além disso, ele fornece o nível de desempenho básico como uma porcentagem do desempenho total do núcleo (usando um único vCPU).

CPUcréditos ganhos por hora	Máximo de créditos obtidos que podem ser acumulados*	vCPUs	Desempenho básico por v CPU	Memória (GiB)	Performance 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



CPUcréditos ganhos por hora	Máximo de créditos obtidos que podem ser acumulados*	vCPUs	Desempenho básico por v CPU	Memória (GiB)	Performance 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
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 básico na tabela é por v. CPU Alguns tamanhos de nós que têm mais de um CPU v. Para isso, calcule a CPU utilização da linha de base para o nó multiplicando a CPU porcentagem v pelo número de. vCPUs

As seguintes métricas CPU de crédito estão disponíveis para instâncias de desempenho intermitente T3 e T4G:

**Note**

Essas métricas não estão disponíveis para instâncias expansíveis T2.

- CPUCreditUsage
- CPUCreditBalance

Para obter mais informações sobre essas métricas, consulte [Métricas CPU de crédito](#).

Além disso, saiba estes detalhes:

- Por padrão, todos os tipos de nós da geração atual são criados em uma nuvem privada virtual (VPC) baseada na AmazonVPC.
- Os arquivos OSS somente para anexar do Redis (AOF) não são compatíveis com instâncias T2. Variáveis de OSS configuração do Redis `appendonly` e `appendfsync` não são compatíveis com a OSS versão 2.8.22 e posterior do Redis.

## Informações relacionadas

- [Características e detalhes ElastiCache do produto Amazon](#)
- [Parâmetros específicos do tipo de nó do Memcached para Memcached](#)
- [Parâmetros Valkey e Redis OSS](#)
- [Criptografia em trânsito \(TLS\)](#)

## Reinicializando nós

Algumas mudanças exigem que um cluster Redis OSS ou Memcached 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

- [Reinicializando os OSS nós do Redis \(somente o modo de cluster desativado\)](#)
- [Reinicializando um cluster para Memcached](#)

## Reinicializando os OSS nós do Redis (somente o modo de cluster desativado)

Para clusters Valkey ou Redis OSS (modo de cluster desativado), os parâmetros nos grupos de parâmetros que são aplicados somente após a reinicialização são:

- activerehashing
- bancos de dados

Os nós do Redis só podem ser atualizados por meio do ElastiCache console. Você só pode reinicializar um único nó de cada vez. Para reinicializar vários nós, você deve repetir o processo para cada nó.

### Alterações nos parâmetros Valkey ou Redis OSS (modo de cluster ativado)

Se você fizer alterações nos seguintes parâmetros em um cluster Valkey ou Redis OSS (modo de cluster ativado), 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 Valkey ou Redis OSS (modo de cluster ativado). 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.

## Usando o AWS Management Console

Você pode reinicializar um nó usando o ElastiCache console.

Para reinicializar um nó (console)

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

2. Na lista no canto superior direito, escolha a AWS região que se aplica.
3. No painel de navegação esquerdo, escolha Redis OSS.

Uma lista de clusters executando o Redis é OSS exibida.

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.

## Reinicializando um cluster para Memcached

Quando você reinicializa um cluster Memcached, o cluster libera todos os dados e reinicia o 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 ElastiCache console AWS CLI, o ou o ElastiCache API. Se você usa o ElastiCache console, o AWS CLI ou o ElastiCache API, você só pode iniciar a reinicialização de um único cluster. Para reinicializar vários clusters, você deve repetir o processo ou a operação.

### Usando o AWS Management Console

Você pode reinicializar um cluster usando o ElastiCache console.

#### Para reinicializar um cluster (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região na qual você está interessado.
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 reiniciando 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).

### Usando o AWS CLI

Para reinicializar um cluster (AWS CLI), use a `reboot-cache-cluster` CLI operação.

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](#).

## Usando o ElastiCache API

Para reinicializar um cluster usando o ElastiCache API, use a `RebootCacheCluster` ação.

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](#).

## Substituindo nós (Valkey e RedisOSS)

A Amazon ElastiCache frequentemente atualiza sua frota com patches e upgrades aplicados às instâncias sem problemas. No entanto, de tempos em tempos, precisamos reiniciar seus ElastiCache nós 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 pela 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 tomar ao ElastiCache programar a substituição de um de seus nós Valkey ou RedisOSS. Para agilizar a busca das informações de que você precisa para sua situação, escolha no menu a seguir.

- [Do nothing](#)— Deixe a Amazon ElastiCache substituir o nó conforme programado.
- [Change your maintenance window](#): alterar sua janela de manutenção para um melhor momento.
- Configurações do Valkey ou Redis OSS (modo de cluster ativado)
  - [Replace the only node in any Valkey or Redis OSS cluster](#)— Um procedimento para substituir um nó em um OSS cluster Valkey ou Redis 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 Valkey ou Redis aumentando e diminuindo a contagem de réplicas sem tempo de inatividade do OSS 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 Valkey ou Redis OSS (modo de cluster ativado) por meio da expansão horizontal e da expansão.

- Configurações do Valkey ou Redis OSS (modo de cluster desativado)
  - [Replace the only node in any Valkey or Redis OSS cluster](#)— Procedimento para substituir qualquer nó em um OSS cluster Valkey ou Redis 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 Valkey ou Redis aumentando e diminuindo a contagem de réplicas sem tempo de inatividade do OSS cluster.
  - [Replace a node in a Valkey or Redis OSS \(cluster mode disabled\) cluster](#)— Procedimento para substituir um nó em um cluster Valkey ou Redis OSS (modo de cluster desativado) 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 Valkey ou Redis OSS (modo de cluster desativado).
  - [Replace a Valkey or Redis OSS \(cluster mode disabled\) primary node](#)— Um procedimento para substituir manualmente o nó primário em um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado).

#### Opções de substituição de OSS nós Valkey e Redis

- Não fazer nada — Se você não fizer nada, ElastiCache substituirá o nó conforme programado.

Para configurações sem cluster com failover automático ativado, os clusters no Valkey 7.2 e superior e no Redis OSS 5.0.6 e superior são substituídos enquanto o cluster continua on-line e atende às solicitações de gravação recebidas. Para clusters habilitados para failover automático no Redis OSS 4.0.10 ou inferior, você pode notar uma breve interrupção de gravação de até alguns segundos associada às atualizações. DNS

Se o nó for membro de um cluster habilitado para failover automático, ElastiCache com Valkey ou Redis, OSS fornece maior disponibilidade durante patches, atualizações e outras substituições de nós relacionadas à manutenção.

Para ElastiCache as configurações do Valkey ou do Redis OSS Cluster configuradas para uso ElastiCache com clientes Valkey ou Redis OSS Cluster, a substituição agora é concluída enquanto o cluster atende às solicitações de gravação recebidas.



Para configurações sem cluster com failover automático ativado, os clusters no Valkey 7.2 e superior e no Redis OSS 5.0.6 e superior são substituídos enquanto o cluster continua on-line e atende às solicitações de gravação recebidas. Para clusters habilitados para failover automático no Redis OSS 4.0.10 ou inferior, você pode notar uma breve interrupção de gravação de até alguns segundos associada às atualizações. DNS

Se o nó for independente, a Amazon ElastiCache primeiro inicia um nó substituto e depois sincroniza a partir do nó existente. O nó existente não está disponível para solicitações de serviço durante esse período. Quando a sincronização é concluída, o nó existente é encerrado e o novo nó toma seu lugar. ElastiCache faz o possível para reter seus dados durante essa operação.

- **Altere sua janela de manutenção** — Para eventos de manutenção programados, você recebe um e-mail ou um evento de notificação de 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 as informações a seguir.
  - [Modificando um cluster ElastiCache](#)
  - [Modificação de um grupo de replicação](#)

#### Note

A capacidade de alterar a janela de substituição movendo a janela de manutenção só está disponível quando a ElastiCache notificação 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 obter instruções, consulte [Gerenciando a manutenção ElastiCache do cluster](#).

- Substitua o único nó em qualquer OSS cluster Valkey ou Redis — Se o cluster não tiver nenhuma réplica de leitura, você poderá 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 obter instruções, consulte [Realização de backups manuais](#).
  2. Crie um novo cluster propagando-o a partir do snapshot. Para obter instruções, consulte [Restauração de um backup para um novo cache](#).
  3. Exclua o cluster com o nó programado para substituição. Para obter instruções, consulte [Excluindo um cluster no ElastiCache](#).
  4. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
- Substitua um nó de réplica em qualquer cluster Valkey ou Redis — Para substituir um OSS 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 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.
- Substitua qualquer nó em um fragmento Valkey ou Redis OSS (modo de cluster ativado) — Para substituir o nó em um cluster sem 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 Valkey ou Redis OSS (modo de cluster ativado)

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 Valkey ou Redis OSS (modo de cluster desativado) — Se o cluster for um cluster Valkey ou Redis OSS (modo de cluster desativado) 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 obter instruções, consulte [Para adicionar replicação a um OSS cluster Valkey ou Redis sem fragmentos](#).
2. Adicione uma réplica de leitura ao cluster. Para obter instruções, consulte [Para adicionar nós a um ElastiCache cluster \(console\)](#).
3. Promova a réplica de leitura recém-criada para primário. Para obter instruções, consulte [Promovendo uma réplica de leitura para primária, para grupos de replicação Valkey ou Redis OSS \(modo de cluster desativado\)](#).

4. Exclua o nó agendado para substituição. Para obter instruções, consulte [Removendo nós de um ElastiCache cluster](#).
  5. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
- Substitua uma réplica de leitura Valkey ou Redis OSS (modo de cluster desativado) — 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 obter instruções, consulte [Modificação de um grupo de replicação](#).

Para substituir uma réplica de leitura do Valkey ou do Redis OSS (modo de cluster desativado)

1. Exclua a réplica programada para substituição. Para obter 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 obter instruções, consulte:
    - [Aumento do número de réplicas em um fragmento](#)
    - [Adicionar uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#)
  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 obter instruções, consulte [Habilitar Multi-AZ](#).
- Substitua um nó primário Valkey ou Redis OSS (modo de cluster desativado) — Se o nó for o nó primário, primeiro promova uma réplica de leitura para primária. 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 obter instruções, consulte [Modificação de um grupo de replicação](#).

Para substituir um nó primário Valkey ou Redis OSS (modo de cluster desativado)

1. Promover uma réplica de leitura para primário. Para obter instruções, consulte [Promovendo uma réplica de leitura para primária, para grupos de replicação Valkey ou Redis OSS \(modo de cluster desativado\)](#).
2. Exclua o nó programado para substituição (o primário antigo). Para obter 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 obter instruções, consulte [Adicionar uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#).

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 obter instruções, consulte [Habilitar Multi-AZ](#).

## Substituindo nós (Memcached)

A Amazon ElastiCache (Memcached) frequentemente atualiza sua frota com patches e upgrades aplicados às instâncias sem problemas. No entanto, de tempos em tempos, precisamos reiniciar seus nós ElastiCache (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 pela 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 tomar ao ElastiCache programar a substituição de um dos seus nós do Memcached.

- Não fazer nada — Se você não fizer nada, ElastiCache substituirá o nó conforme programado. Quando substitui ElastiCache automaticamente o nó por um novo nó, o novo nó fica inicialmente vazio.
- Altere sua janela de manutenção — Para eventos de manutenção programados, você recebe um e-mail ou um evento de notificação de 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 [Modificando um cluster ElastiCache](#).

### Note

A capacidade de alterar a janela de substituição movendo a janela de manutenção só está disponível quando a ElastiCache notificação 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 obter instruções, consulte [Gerenciando a manutenção ElastiCache do cluster](#).

- 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 obter instruções, consulte [Removendo nós de um ElastiCache cluster](#).
2. Adicione um novo nó ao cluster. Para obter instruções, consulte [Adicionando 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 ElastiCache nós pode ser uma forma 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 sobre preços, consulte [Amazon ElastiCache Pricing](#).

## 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 sobre preços, consulte [Amazon ElastiCache Pricing](#).

AWS A região, o tipo de nó e a duração do período devem ser escolhidos na compra e não podem ser alterados posteriormente.

Você pode usar o AWS Management Console AWS CLI, o ou o ElastiCache API para listar e comprar ofertas de nós reservados disponíveis.

Para obter mais informações sobre nós reservados, consulte [Amazon ElastiCache Reserved Nodes](#).



## Ofertas de nós reservados padrão

Ao comprar uma instância de nó reservado (RI) na Amazon ElastiCache, você pode adquirir o compromisso de obter uma taxa de desconto em um tipo de instância de nó e AWS região específicos durante a duração da instância de nó reservado. Para usar uma instância de nó ElastiCache reservado da Amazon, você cria uma nova instância de ElastiCache nó, da mesma forma que faria com uma instância sob demanda.

Se as especificações da nova instância do nó de reserva corresponderem a uma instância de nó reservado existente da sua conta, você será cobrado de acordo com a taxa de desconto oferecida para a instância do nó reservado. Caso contrário, uma taxa sob demanda será cobrada para a instância do nó. Esses padrões RIs estão disponíveis a partir das famílias de instâncias R5 e M5.

### Note

Todos os tipos de oferta discutidos a seguir estão disponíveis em prazos de um e três anos.

## Tipos de oferta

Nenhuma RI antecipada fornece acesso a uma ElastiCache instância reservada sem exigir um pagamento adiantado. Sua ElastiCache instância reservada No Upfront cobra uma taxa horária com desconto para cada hora dentro do prazo, independentemente do uso.

A RI antecipada parcial exige que uma parte da ElastiCache instância reservada 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 incorrem 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. Quando você compra um nó reservado, uma coisa que você especifica é o tipo de nó, por exemplo `cache.r6g.xlarge`. Para obter mais informações sobre os tipos de nós, consulte [Amazon ElastiCache Pricing](#).

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. 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 sem custo adicional. AWS

### Atualização de nós reservados do Redis OSS para o Valkey

Com o lançamento do Valkey in ElastiCache, agora você pode aplicar seu desconto em nós OSS reservados do Redis ao mecanismo de cache do Valkey. Você pode fazer o upgrade do Redis OSS para o Valkey enquanto ainda se beneficia dos contratos e reservas existentes. Além de poder aplicar seus benefícios na família e no mecanismo de nós de cache, você pode até receber mais valor incremental. O Valkey tem um preço de 20% de desconto em relação ao Redis eOSS, com a flexibilidade de nós reservados, você pode usar seus nós OSS reservados do Redis para cobrir 20% a mais de nós em execução do Valkey.

Para calcular a taxa de desconto, cada combinação de ElastiCache nó e motor 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. OSSOs nós reservados do Redis também podem ser aplicados em todos os mecanismos para cobrir os nós do Valkey em execução. 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 OSS reservado do Redis cubra mais nós do Valkey.

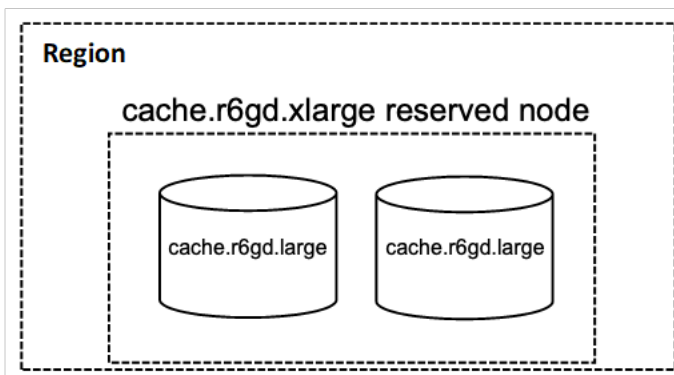
Por exemplo, digamos que você tenha comprado um nó reservado para um `cache.r7g.4xlarge` para o OSS mecanismo Redis (32 unidades) e esteja executando um nó Redis `cache.r7g.4xlarge` (32 unidades). OSS Se você atualizar o nó para Valkey, o fator de normalização do nó em execução cai para 25,6 unidades, e seu nó reservado existente fornece 6,4 unidades adicionais para usar em qualquer outro nó Valkey ou Redis em execução na família OSS `cache.r7g` na região. Você pode usar isso para cobrir 25% de outro nó do Valkey `cache.r7g.4xlarge` na conta (25,6 unidades) ou 100% de um nó do Valkey `cache.r7g.xlarge` (6,4 unidades).

## Comparando o uso com unidades normalizadas

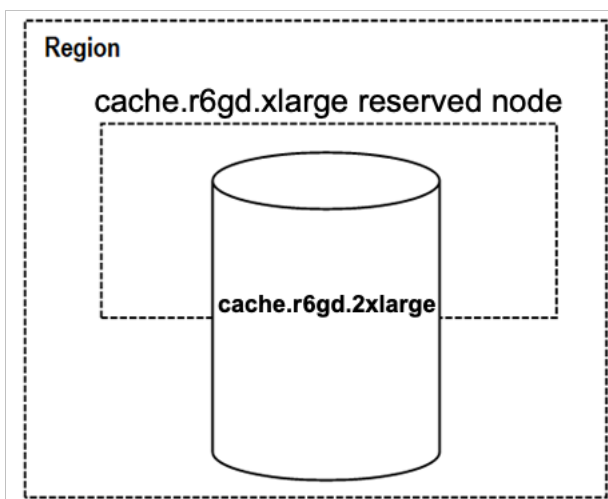
Você pode comparar o uso de diferentes tamanhos de nós reservados usando unidades normalizadas. Por exemplo, uma hora de uso em dois nós cache.r6g.4xlarge é equivalente a 16 horas de uso em um cache.r6g.large. A tabela a seguir mostra o número de unidades normalizadas para cada tamanho de nó:

Tamanho do nó	Unidades normalizadas (RedisOSS, Memcached)	Unidades normalizadas (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 em sua conta na mesma região. AWS Nesse caso, o benefício de faturamento é aplicado integralmente a ambos os nós.



Como alternativa, se você tiver uma instância cache.r6gd.2xlarge em execução na sua conta na mesma AWS região, 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 [Amazon ElastiCache Pricing](#).

## 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 preços e informações sobre as ofertas de nós reservados disponíveis usando o AWS Management Console AWS CLI, e. ElastiCache API

### Tópicos

- [Obtenção de informações sobre ofertas de nós reservados \(console\)](#)
- [Obtenção de informações sobre ofertas de nós reservados \(AWS CLI\)](#)
- [Obter informações sobre ofertas de nós reservados \(\) ElastiCache API](#)

### Obtenção de informações sobre ofertas de nós reservados (console)

Para obter preços e outras informações sobre as ofertas de cluster reservado disponíveis usando o AWS Management Console, use o procedimento a seguir.

Para obter informações sobre ofertas de nós reservados disponíveis

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione Reserved Nodes.
3. Escolha Purchase Reserved Nodes (Comprar nós reservados).
4. Para Engine, escolha Valkey, Redis ou OSS Memcached.
5. Para determinar as ofertas disponíveis, faça seleções nas seguintes opções:
  - Node Type
  - 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 as ofertas de nós reservados disponíveis para Valkey ou RedisOSS, digite o seguinte comando em um prompt de comando:

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Essa operação produz uma saída semelhante à seguinte (JSONformato):

```
{
  "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 as ofertas de nós reservados disponíveis para o 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.

Obter informações sobre ofertas de nós reservados () ElastiCache API

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 ElastiCache API Referência.

## Compra de um nó reservado

Os exemplos a seguir mostram como comprar uma oferta de nó reservado usando o AWS Management Console, o AWS CLI, o e ElastiCache API.

### Important

Seguir os exemplos desta seção gera cobranças em sua AWS conta que você não pode reverter.

### Tópicos

- [Compra de um nó reservado \(console\)](#)
- [Compra de um nó reservado \(AWS CLI\)](#)
- [Compra de um nó reservado \(ElastiCache API\)](#)

### 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 AWS Management Console para comprar a oferta de nó reservado por meio do ID da oferta.

### Para comprar nós reservados

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 o link Reserved nodes (Nós reservados).
3. Escolha o botão Purchase reserved nodes (Comprar nós reservados).
4. Em Engine, escolha Valkey, Redis ou OSS Memcached.
5. Para determinar as ofertas disponíveis, faça seleções nas seguintes opções:
  - Node Type
  - 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-offerting](#) na AWS CLI Referência.

### Compra de um nó reservado (ElastiCache API)

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 ElastiCache API Referência.

## Obtenção de informações sobre seus nós reservados

Você pode obter informações sobre os nós reservados que você comprou usando o AWS Management Console, o AWS CLI, ou a ElastiCache API.

### Tópicos

- [Obtenção de informações sobre seus nós reservados \(console\)](#)
- [Obtenção de informações sobre seus nós reservados \(AWS CLI\)](#)
- [Obtendo informações sobre seus nós reservados \(ElastiCache API\)](#)

### Obtenção de informações sobre seus nós reservados (console)

O procedimento a seguir descreve como usar o AWS Management Console para obter informações sobre os nós reservados que você adquiriu.

Para obter informações sobre seus nós reservados comprados

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 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 AWS conta, digite o seguinte comando em um prompt de comando:

```
aws elasticache describe-reserved-cache-nodes
```

Essa operação produz uma saída semelhante à seguinte (JSONformato):

```
{
  "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 AWS CLI Referência.

Obtendo informações sobre seus nós reservados (ElastiCache API)

Para obter informações sobre nós reservados para sua AWS conta, chame a DescribeReservedCacheNodes operação.

### 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 ElastiCache API Referência.

## 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 da geração anterior, 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 nós compatíveis e dos tipos de nós da geração anterior em ElastiCache, consulte [Tipos de nó compatíveis](#).

## Migração de nós em um cluster Valkey ou Redis OSS

O procedimento a seguir descreve como migrar seu tipo de nó de OSS cluster Valkey ou Redis usando o console. ElastiCache Durante esse processo, seu OSS cluster Valkey ou Redis continuará atendendo às solicitações com o mínimo de tempo de inatividade. 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 de cluster desativado (nó único) pode durar aproximadamente 60 segundos, principalmente devido à DNS propagação.
- O modo de cluster desativado (com nó de réplica) pode levar aproximadamente 1 segundo para clusters que executam Valkey 7.2 e superior ou Redis OSS 5.0.6 e superior. Todas as versões inferiores podem durar aproximadamente 10 segundos.
- O modo cluster habilitado pode ver aproximadamente 1 segundo.

Para modificar um tipo de nó de OSS cluster Valkey ou Redis usando o console:

1. Faça login no console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters Redis OSS.
3. Na lista de clusters, escolha o cluster que você deseja migrar.
4. Escolha Ação e 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 OSS cluster Valkey ou Redis usando: AWS CLI

Use o [modify-replication-group](#) API 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.

### Fixando ou removendo `restore-failed-node` (s)

O procedimento a seguir descreve como corrigir ou remover nós com falha de restauração do seu cluster Valkey ou Redis. OSS Para saber mais sobre como os ElastiCache nós entram em um estado de falha na restauração, consulte [Visualizando o status ElastiCache do nó](#). Recomendamos primeiro remover todos os nós em um estado de falha na restauração, depois migrar os nós restantes da geração anterior no ElastiCache cluster para um tipo de nó de nova geração e, por fim, adicionar novamente o número necessário de nós.

Para remover o nó com falha de restauração (console):

1. Faça login no console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters 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 que falharam na restauração do seu ElastiCache cluster, agora você pode migrar para um tipo de geração mais recente. Para obter mais informações, consulte acima em [Migração de nós em um cluster Valkey ou Redis OSS](#).

Para adicionar novamente nós ao seu ElastiCache cluster, consulte [Adicionando nós a um ElastiCache cluster](#).

## Migração de nós em um cluster do Memcached

Para migrar ElastiCache (Memcached) para um tipo de nó diferente, você deve criar um novo cluster, que sempre começa vazio para que seu aplicativo possa preencher.

Para migrar seu tipo de nó de cluster ElastiCache (Memcached) usando o console: 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 [Encontrando os endpoints de um cluster \(console\) \(Memcached\)](#)
- Exclua o cluster antigo. Para obter mais informações, consulte [Excluindo um cluster no ElastiCache](#)

## Gerenciando clusters em ElastiCache

Um cluster é uma coleção de um ou mais nós de cache, todos executando uma instância do software de motor Valkey, Redis OSS ou Memcached. Ao criar um cluster, especifique o mecanismo e a versão para que todos os nós usem.

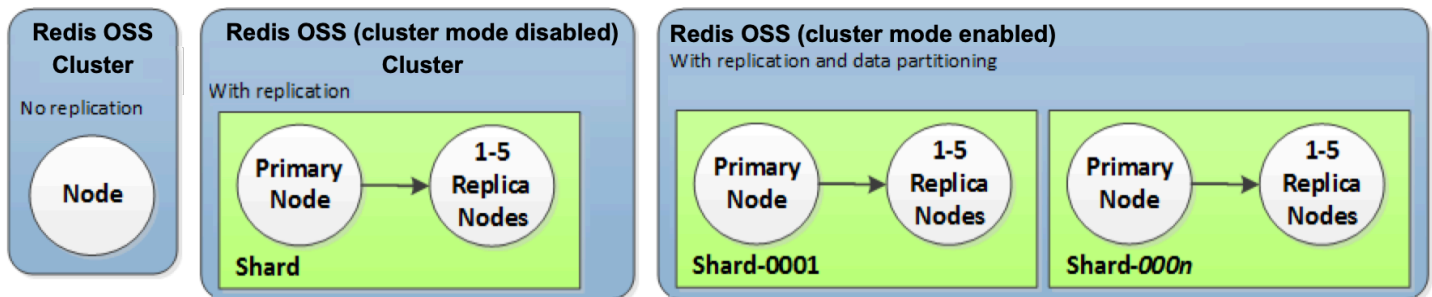
### Clusters Valkey e Redis OSS

O diagrama a seguir ilustra um cluster típico do Valkey ou do RedisOSS. Esses clusters podem conter um único nó ou até seis nós dentro de um fragmento (API/CLI: grupo de nós), um cluster Valkey ou Redis de nó único OSS (modo de cluster desativado) não tem fragmento e um cluster Valkey ou Redis de vários nós OSS (modo de cluster desativado) tem um único fragmento. Os clusters Valkey ou Redis OSS (modo de cluster ativado) podem ter até 500 fragmentos, com seus dados particionados entre os 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 superior ou Redis OSS 5.0.6 e 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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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).

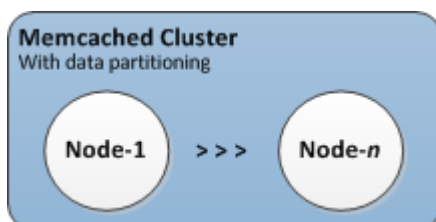
Quando você tem vários nós em um OSS fragmento Valkey ou Redis, um dos nós é um nó primário de leitura/gravação. Todos os outros nós no estilhaço são réplicas somente leitura.

Os OSS clusters típicos do Valkey ou do Redis têm a seguinte aparência.



## Clusters Memcached

Os clusters típicos do Memcached têm a seguinte aparência. Os clusters do Memcached contêm de 1 a 60 nós, através dos quais você particiona horizontalmente seus dados.



## Operações do ElastiCache para Valkey, OSS Redis e Memcached

A maioria das ElastiCache operações é executada 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 ElastiCache API e AWS CLI. O identificador do cluster deve ser exclusivo para esse cliente em uma AWS região.

ElastiCache suporta várias versões do motor. A menos que você tenha motivos específicos, recomendamos usar a versão mais recente.

ElastiCache os clusters são projetados para serem acessados usando uma EC2 instância da Amazon. Se você iniciar seu cluster em uma nuvem privada virtual (VPC) baseada no VPC serviço da Amazon, poderá acessá-lo de fora AWS. Para obter mais informações, consulte [Acessando ElastiCache recursos de fora AWS](#).

Para obter uma lista das versões compatíveis [Mecanismos e versões compatíveis](#), consulte [Versões do Redis OSS suportadas](#), [Versões suportadas ElastiCache \(Memcached\)](#) e.

## Escolhendo um tipo de rede em ElastiCache

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4eIPv6), 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 o Valkey 7.2 e versões posteriores, ou o OSS mecanismo Redis versão 6.2 e 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 mecanismo Memcached versão 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 AmazonVPC, 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 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.

## Usar pilha dupla

Ao usar ElastiCache (RedisOSS) 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 de cache com ElastiCache (Memcached) e escolher pilha dupla 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 IPIPv6, 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 [Identifique automaticamente os nós em 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.

## TLScusters de pilha ElastiCache dupla habilitados

Quando TLS está habilitada para ElastiCache clusters, as funções de descoberta de `clusterscluster slots`, `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 TLS handshake. Isso significa que os clientes não serão afetados pelo parâmetro de descoberta de IP. Para clusters TLS habilitados, o parâmetro IP Discovery não tem efeito no protocolo IP preferido. Em vez disso, o protocolo IP usado será determinado pelo protocolo IP que o cliente prefere ao resolver DNS nomes de host.

Para obter exemplos de como configurar uma preferência de protocolo IP ao resolver DNS nomes de host, consulte [TLScusters de pilha ElastiCache dupla habilitados](#)

## Usando o AWS Management Console (Valkey e RedisOSS)

Ao criar um cluster usando o AWS Management Console, 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 obter mais informações, consulte [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#) ou [Criação de um Valkey ou Redis OSS \(modo de cluster desativado\) \(Console\)](#).

Ao criar um grupo de replicação usando o AWS Management Console, 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 [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) do zero](#) ou [Criando um grupo de replicação no Valkey ou no Redis OSS \(modo de cluster ativado\) do zero](#).

## Usando o AWS Management Console (Memcached)

Ao criar um cluster de cache usando o AWS Management Console, 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 \(console\)](#).

## Usando o CLI com Valkey, Redis OSS ou Memcached

### Redis OSS

Ao criar um cluster de cache com Valkey ou Redis OSS usando o CLI, você usa o [create-cache-cluster](#) comando e especifica os parâmetros NetworkType e 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 o CLI, você usa o [create-replication-group](#) comando e especifica os IPDiscovery parâmetros NetworkType 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 o CLI, você usa o [create-replication-group](#) comando e especifica os IPDiscovery parâmetros NetworkType 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 \  
  --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 o CLI, você usa o [create-replication-group](#) comando e especifica os IPDiscovery parâmetros NetworkType 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 de cache com o Memcached usando o CLI, você usa o [create-cache-cluster](#) comando e especifica os parâmetros NetworkType e 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
```

## Identifique automaticamente os nós em 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 de cache e iniciarem e manterem conexões com todos esses nós.

### Note

A descoberta automática é adicionada para clusters de cache executados no Amazon ElastiCache Memcached. O Auto Discovery não está disponível para motores Valkey ou RedisOSS.

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](#)
- [Conectando-se manualmente aos nós de cache do Memached](#)
- [Adicionando o Auto Discovery à sua biblioteca de cliente do Memcached](#)
- [ElastiCache clientes com descoberta automática](#)

## Benefícios da descoberta automática com o Memcached

Ao usar o Memcached, o Auto Discovery oferece os seguintes benefícios:

- Quando você aumenta o número de nós em um cluster de cache, 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 de cache ElastiCache do Memcached. Você não precisa reinicializar nenhum dos seus nós de cache para usar esse recurso.

## Como a Descoberta automática funciona

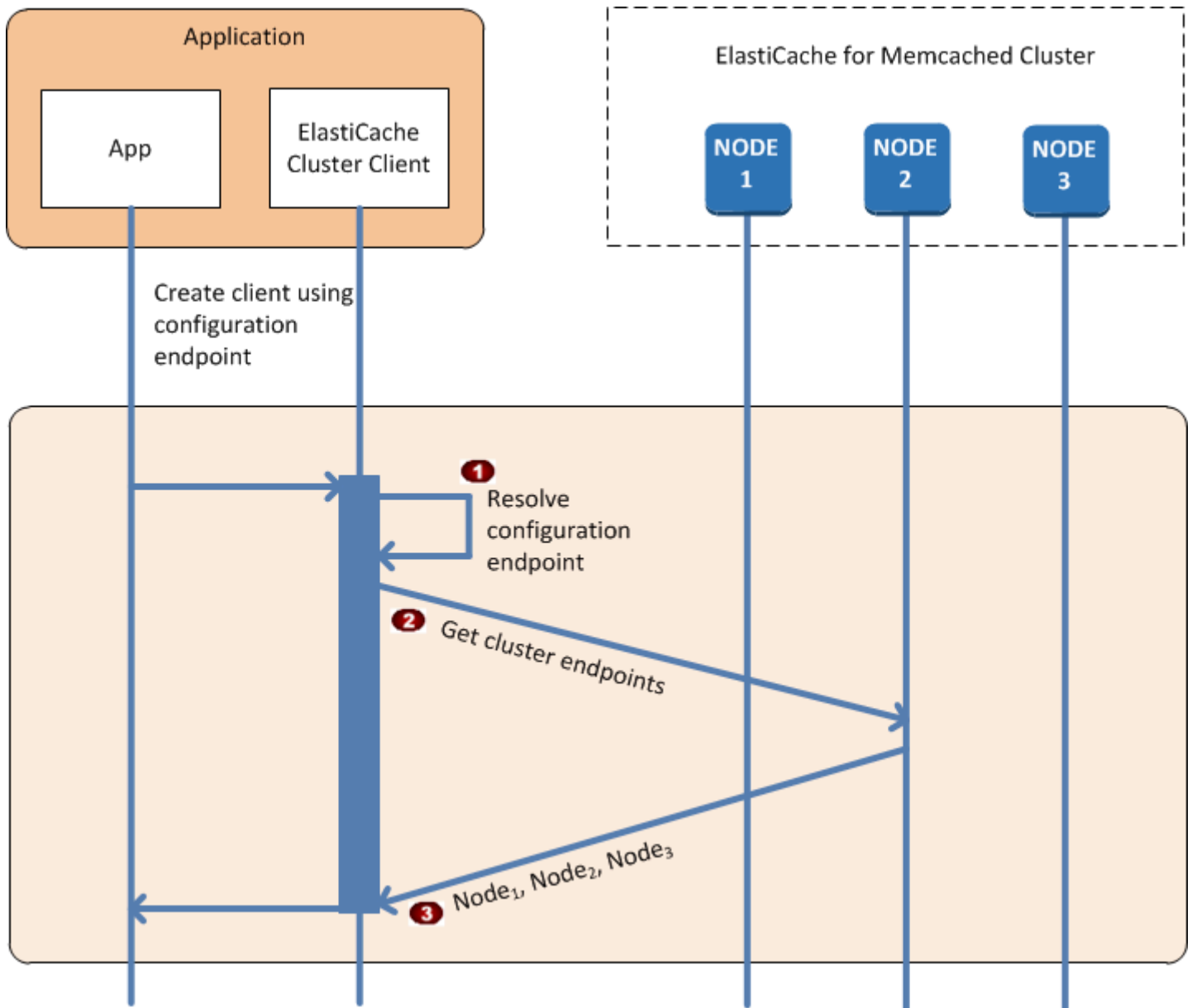
### Tópicos

- [Conexão com nós de cache](#)
- [Operações normais de cluster](#)
- [Outras operações do](#)

Esta seção descreve como os aplicativos cliente usam o ElastiCache Cluster Client 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 do endpoint de DNS configuração. Como o endpoint de configuração mantém CNAME entradas para todos os nós de cache, o DNS nome 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.

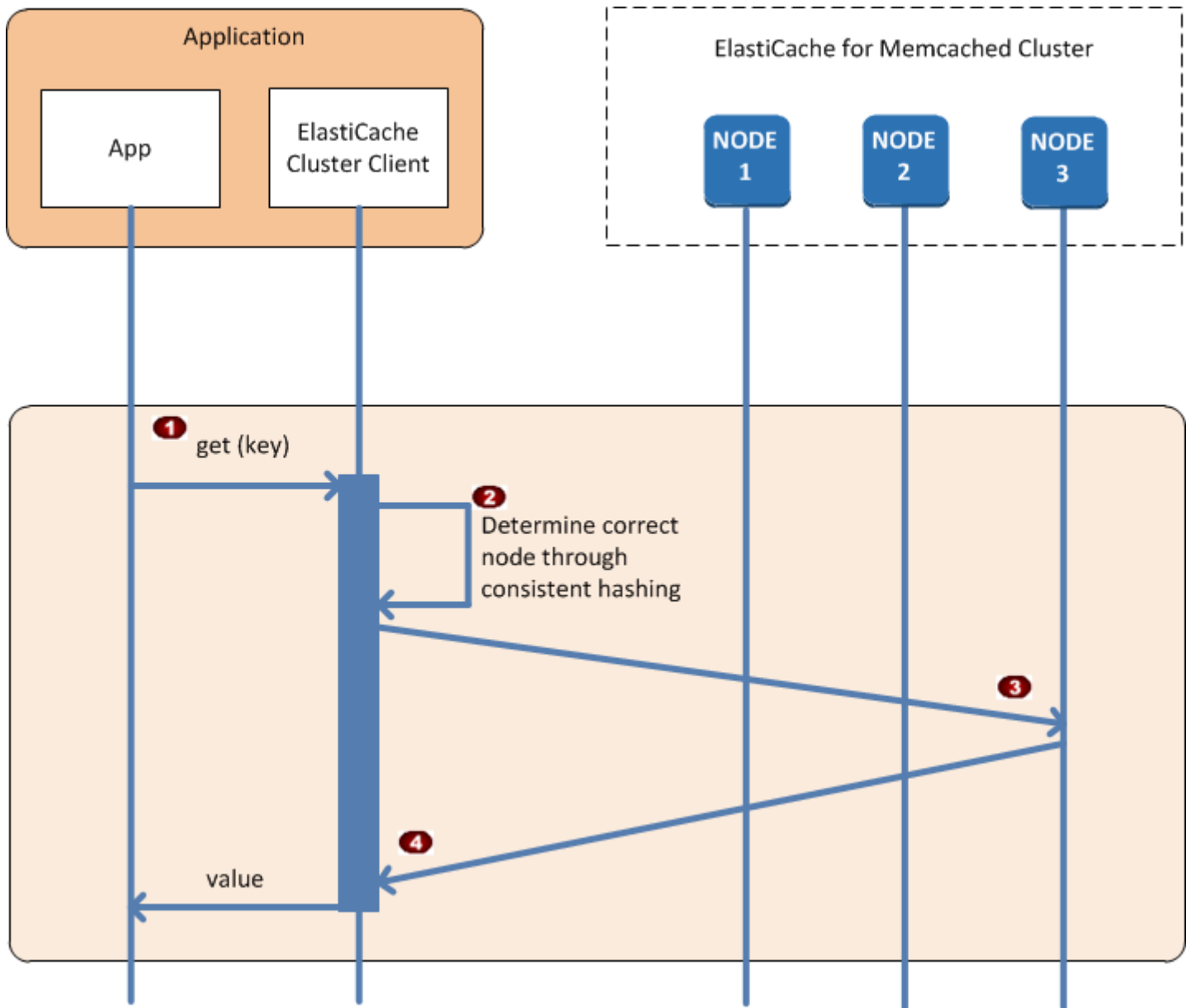
 Note

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 o aplicativo se conecta a todos os nós de cache, o ElastiCache Cluster Client determina quais nós devem armazenar itens de dados individuais e quais nós devem ser consultados para esses itens de dados posteriormente. 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 do

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, ElastiCache derruba esse nó e ativa um substituto. 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 o Auto Discovery com ElastiCache (Memcached), siga estas etapas:

- [Obtenha o endpoint de configuração](#)
- [Baixe o ElastiCache Cluster Client](#)
- [Modifique seu programa de inscrição](#)

### Obtenha o endpoint de configuração

Para se conectar a um cluster, os programas cliente devem conhecer o endpoint de configuração de cluster. Consulte o tópico [Encontrando os 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 Encontrando endpoints usando o for AWS CLI 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
```

Essa operação produz uma saída semelhante à seguinte (JSONformato):

```
{  
  "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 ElastiCache Cluster Client

Para aproveitar as vantagens do Auto Discovery, os programas cliente devem usar o ElastiCacheCluster Client. O ElastiCache Cluster Client está disponível para JavaPHP, e. NETe contém toda a lógica necessária para descobrir e se conectar a todos os seus nós de cache.

### Para baixar o ElastiCache Cluster Client

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No ElastiCache console, escolha ElastiCache Cluster Client e escolha Download.

O código-fonte do ElastiCache Cluster Client for Java está disponível em <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java>. Essa biblioteca se baseia no popular cliente Spymemcached. O ElastiCache Cluster Client 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 ElastiCache Cluster ClientPHP, primeiro você precisará instalá-lo em sua EC2 instância da Amazon. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

Para um cliente TLS compatível, baixe o binário com a PHP versão 7.4 ou superior.

Para usar o ElastiCache Cluster Client para. NET, primeiro você precisará instalá-lo em sua EC2 instância da Amazon. Para obter mais informações, consulte [Instalando o cliente de ElastiCache cluster para. NET](#).

## Modifique seu programa de inscrição

Modifique seu programa aplicativo para que ele use a Descoberta automática. As seções a seguir mostram como usar o ElastiCache Cluster Client para JavaPHP, 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" nele.

```
"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.

## Usando o ElastiCache Cluster Client para Java

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 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 PHP, primeiro você precisará instalá-lo em sua EC2 instância da Amazon. Para ter 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 cache cluster up or down in number  
 * of nodes  
 * without requiring any changes to the PHP application.  
 *  
 * By default the Memcached instances are destroyed at the end of the request.  
 * To create an instance that persists between requests,  
 * use persistent_id to specify a unique ID for the instance.  
 * All instances created with the same persistent_id will share the same connection. */
```

```
* See http://php.net/manual/en/memcached.construct.php for more information.
*/
$dynamic_client = new Memcached('persistent-id');
$dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
$dynamic_client->addServer($server_endpoint, $server_port);

/**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
$dynamic_client->set('key', 'value', 60);

/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
$static_client = new Memcached('persistent-id');
$static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
$static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
$static_client->set('key', 'value');
?>
```

Para ver um exemplo de como usar o ElastiCache Cluster Client com TLS ativado, consulte [Usando criptografia em trânsito com PHP e Memcached](#).

Usando o ElastiCache Cluster Client para .NET

#### Note

ElastiCache A. NETo cliente de cluster foi descontinuado em maio de 2022.

.NETO cliente para ElastiCache é de código aberto em <https://github.com/awslabs/elasticache-cluster-config-net>.



. NETos aplicativos normalmente obtêm suas configurações do 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 C# 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.

```
// *****
// Sample C# code to show how to integrate with the Amazon ElastiCcache Auto Discovery
// feature.

using System;

using Amazon.ElastiCacheCluster;

using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo {

    public static void Main(String[] args) {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);

        // Store the data for 3600 seconds (1hour) in the cluster.
```

```
// The client will decide which cache host will store this item.  
memClient.Store(StoreMode.Set, 3600, "This is the data value.");  
  
} // end Main  
  
} // end class DotNetAutoDiscoverDemo
```

## Conectando-se manualmente aos nós de cache do Memcached

Se o seu programa cliente não usar o Auto Discovery, 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 cache 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 de cache adicionando ou removendo nós, precisará atualizar a lista de nós no código do cliente.

## Adicionando o Auto Discovery à sua biblioteca de cliente do Memcached

As informações de configuração do Auto Discovery são armazenadas de forma redundante em cada nó do cluster de cache 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 ElastiCache Cluster Client. 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 documentação a seguir se refere ao ASCII protocolo; no entanto, o `config` comando oferece suporte a ambos ASCII e ao 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

Nome	Descrição	Obrigatório
sub-command	O subcomando usado para interagir com um nó de cache. Para a Descoberta automática, esse subcomando é <code>get</code> .	Sim

Nome	Descrição	Obrigatório
key	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 inferior a 1.4.14

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 de cache, 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 de cache 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 endereço IP privado CNAME e o endereço IP. Sempre CNAME estará presente; se o endereço IP privado não estiver disponível, ele não será exibido; no entanto, os caracteres de barra vertical "|" ainda serão 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 de cache que estão executando o mecanismo Memcached.

Esta seção discute a instalação e a configuração do e. ElastiCache PHP NETclientes para uso com descoberta automática.

### Tópicos

- [Instalação e compilação de clientes de cluster](#)
- [Configurando clientes ElastiCache](#)

## Instalação e compilação de clientes de cluster

Esta seção aborda a instalação, configuração e compilação do e. PHP NETClientes de cluster de descoberta ElastiCache automática da Amazon.

### Tópicos

- [Instalando o cliente de ElastiCache cluster para .NET](#)
- [Instalação do cliente de cluster do ElastiCache para PHP](#)
- [Compilando o código-fonte para o cliente de ElastiCache cluster para PHP](#)

## Instalando o cliente de ElastiCache cluster para .NET

Você pode encontrar ElastiCache o. NETCódigo do Cluster Client como código aberto em <https://github.com/awslabs/elasticache-cluster-config-net>.

Esta seção descreve como instalar, atualizar e remover o. NETcomponentes para o ElastiCache Cluster Client em EC2 instâncias da Amazon. Para obter mais informações sobre a descoberta automática, consulte [Identifique automaticamente os nós em seu cluster \(Memcached\)](#). Para amostra. NETcódigo para usar o cliente, consulte [Usando o ElastiCache Cluster Client para .NET](#).

### Tópicos

- [Instalando. NET](#)
- [Baixe ElastiCache o. NETcliente de cluster para ElastiCache](#)
- [Instale AWS montagens com NuGet](#)

## Instalando. NET

Você deve ter. NET3.5 ou posterior instalado para usar AWS o. NETSDKpara ElastiCache. Se você não tem. NET3.5 ou posterior, você pode baixar e instalar a versão mais recente em <http://www.microsoft.com/net>.

## Baixe ElastiCache o. NETcliente de cluster para ElastiCache

Para baixar ElastiCache o. NETcliente de cluster

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, clique em ElastiCache Cluster Client.

3. Na lista Baixar clientes do ElastiCache Memcached Cluster, selecione. NETe, em seguida, clique em Baixar.

## Instale AWS montagens com NuGet

NuGet é um sistema de gerenciamento de pacotes para o .NET plataforma. NuGet está ciente das dependências de montagem e instala todos os arquivos necessários automaticamente. NuGet os conjuntos instalados são armazenados com sua solução, em vez de em um local central, como Program Files, por exemplo, para que você possa instalar versões específicas de um aplicativo sem criar problemas de compatibilidade.

### Instalando NuGet

NuGet pode ser instalado a partir da Galeria de Instalação em MSDN; consulte <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Se você estiver usando o Visual Studio 2010 ou posterior, NuGet é instalado automaticamente.

Você pode usar NuGet a partir do Solution Explorer ou do Package Manager Console.

### Usando NuGet do Solution Explorer

Para usar a NuGet partir do Solution Explorer no Visual Studio 2010

1. No menu Ferramentas, selecione Gerenciador de Pacotes de Biblioteca.
2. Clique em Console do Gerenciador de Pacotes.

Para usar a NuGet partir do Solution Explorer no Visual Studio 2012 ou no Visual Studio 2013

1. No menu Tools, selecione NuGet Package Manager.
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 para cada pacote que está disponível NuGet, como os assemblies AWS SDK e AWS.Extensions, consulte o NuGet site em http://www.nuget.org.](http://www.nuget.org) A página de cada pacote inclui



um exemplo de linha de comando para instalar o pacote usando o console e uma lista das versões anteriores do pacote que estão disponíveis noNuGet.

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 [Identifique automaticamente os nós em 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](#)
- [Removendo o cliente PHP de cluster](#)

### Para baixar o pacote de instalação

Para garantir o uso da versão correta do ElastiCache Cluster ClientPHP, você precisará saber qual versão do PHP está instalada na sua EC2 instância da Amazon. Você também precisará saber se sua EC2 instância da Amazon está executando uma versão Linux de 64 ou 32 bits.

Para determinar a PHP versão instalada na sua EC2 instância da Amazon

- No prompt de comando, execute o seguinte comando:

```
php -v
```

A PHP versão 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 sua versão PHP e a do Memcached forem incompatíveis, você receberá uma mensagem de erro parecida com 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 [Compilando o código-fonte para o cliente de ElastiCache cluster para PHP](#).

Para determinar sua EC2 AMI arquitetura Amazon (64 bits ou 32 bits)

1. Faça login no AWS Management Console e abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
2. Na lista de instâncias, clique na sua EC2 instância da Amazon.
3. Na guia Descrição, 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 ElastiCache Cluster Client.

Para baixar o cliente ElastiCache de cluster para PHP

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No ElastiCache console, escolha ElastiCache Cluster Client.
3. Na lista Baixar clientes de cluster do ElastiCache Memcached, escolha o cliente de ElastiCache cluster que corresponda à sua PHP versão e AMI arquitetura e, em seguida, escolha o botão Download.

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 [Removendo o cliente PHP de cluster](#).
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

- [Instalando o PHP 7.x para novos usuários](#)
- [Instalando o PHP 5.x para novos usuários](#)

## Instalando o PHP 7.x para novos usuários

### Tópicos

- [Para instalar o PHP 7 em um servidor Ubuntu 14.04 LTS AMI \(64 bits e 32 bits\)](#)
- [Para instalar o PHP 7 em um Amazon Linux 201609 AMI](#)
- [Para instalar o PHP 7 em um SUSE Linux AMI](#)

## Para instalar o PHP 7 em um servidor Ubuntu 14.04 LTS AMI (64 bits e 32 bits)

1. Execute uma nova instância a partir doAMI.
2. Execute os seguintes comandos:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Instale PHP 7.

```
sudo yum install php70
```

4. Baixe o Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. EXTRACT 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 um Amazon Linux 201609 AMI

1. Execute uma nova instância a partir do AMI.
2. Execute o seguinte comando:

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

3. Instale PHP 7.

```
sudo yum install php70
```

4. Baixe o Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. EXTRACT 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 um SUSE Linux AMI

1. Execute uma nova instância a partir doAMI.
2. Execute o seguinte comando:

```
sudo zypper install gcc
```

3. Instale PHP 7.

```
sudo yum install php70
```

4. Baixe o Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/  
latest-64bit
```

5. EXTRACT 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
```

## Instalando o PHP 5.x para novos usuários

### Tópicos

- [Para instalar PHP 5 em um Amazon Linux AMI 2014.03 \(64 bits e 32 bits\)](#)
- [Para instalar PHP 5 em um Red Hat Enterprise Linux 7.0 AMI \(64 bits e 32 bits\)](#)
- [Para instalar PHP 5 em um servidor Ubuntu 14.04 LTS AMI \(64 bits e 32 bits\)](#)
- [Para instalar PHP 5 para o servidor corporativo SUSE Linux 11 AMI \(64 bits ou 32 bits\)](#)
- [Outras distribuições do Linux](#)

### Para instalar PHP 5 em um Amazon Linux AMI 2014.03 (64 bits e 32 bits)

1. Inicie uma instância do Amazon Linux (64 bits ou 32 bits) e faça login nela.
2. Instale PHP dependências:

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

3. Baixe o php-memcached pacote correto para sua EC2 instância e PHP versão da Amazon. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o php-memcached. Esse URI deve ser o caminho de download do pacote de instalação:

```
sudo pecl install <package download path>
```

Aqui está um exemplo de comando de instalação para Linux PHP 5.4, 64 bits. Neste exemplo, substitua *X.Y.Z* com o número real da versão:

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

**Note**

Use a versão recente do artefato de instalação.

- Com a permissão root/sudo, adicione um novo arquivo chamado `memcached.ini` no `/etc/php.d` diretório 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
```

- Inicie ou reinicie seu servidor Apache.

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

Para instalar PHP 5 em um Red Hat Enterprise Linux 7.0 AMI (64 bits e 32 bits)

- Execute uma instância do Red Hat Enterprise Linux (64 bits ou 32 bits) e faça login nela.
- Instale PHP dependências:

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

- Baixe o `php-memcached` pacote correto para sua EC2 instância e PHP versão da Amazon. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
- Instalar o `php-memcached`. Esse URI deve ser o caminho de download do pacote de instalação:

```
sudo pecl install <package download path>
```

- 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
```

- Inicie ou reinicie seu servidor Apache.



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

Para instalar PHP 5 em um servidor Ubuntu 14.04 LTS AMI (64 bits e 32 bits)

1. Inicie uma instância do Ubuntu Linux (64 bits ou 32 bits) e faça login nela.
2. Instale PHP dependências:

```
sudo apt-get update  
sudo apt-get install gcc g++ php5 php-pear
```

3. Baixe o php-memcached pacote correto para sua EC2 instância e PHP versão da Amazon. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o php-memcached. URI Deve ser o caminho de download do 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 `amazon-elasticache-cluster-client.so` do PHP cliente do `*.tgz` arquivo baixado e copiá-lo para o `/usr/lib/php5/20121212*` diretório.

```
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 nomeado `memcached.ini` no `/etc/php5/cli/conf.d` diretório e insira “`extension=<absolute path to .so>`” no arquivo. `amazon-elasticache-cluster-client`

```
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 PHP 5 para o servidor corporativo SUSE Linux 11 AMI (64 bits ou 32 bits)

1. Inicie uma instância SUSE Linux (64 bits ou 32 bits) e faça login nela.
2. Instale PHP dependências:

```
sudo zypper install gcc php53-devel
```

3. Baixe o php-memcached pacote correto para sua EC2 instância e PHP versão da Amazon. Para obter mais informações, consulte [Para baixar o pacote de instalação](#).
4. Instalar o php-memcached. URIDeve ser o caminho de download do 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 a versão PHP em uso é compatível. Oferecemos suporte às versões 5.3 a 5.5.

## Outras distribuições do Linux

Em alguns sistemas, principalmente o Cent OS7 e o Red Hat Enterprise Linux (RHEL), o 7.1 `libsas12.so.3` foi substituído. `libsas12.so.2` Nesses sistemas, quando você carrega o cliente do ElastiCache cluster, ele tenta e não consegue localizar e carregar `libsas12.so.2`. Para resolver esse problema, crie um link simbólico para `libsas12.so.3`, de forma que, quando o cliente tentar carregar `libsas12.so.2`, ele seja redirecionado para `libsas12.so.3`. O código a seguir cria esse link simbólico.

```
cd /usr/lib64
sudo ln libsas12.so.3 libsas12.so.2
```

## Removendo o cliente PHP de cluster

### Tópicos

- [Removendo uma versão anterior do PHP 7](#)
- [Removendo uma versão anterior do PHP 5](#)

## Removendo uma versão anterior do PHP 7

### Para remover uma versão anterior do PHP 7

1. Remova o `amazon-elasticache-cluster-client.so` arquivo do diretório PHP lib apropriado, conforme indicado anteriormente nas instruções de instalação. Consulte a seção para a sua instalação em [Para usuários que já possuem 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
```

## Removendo 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.

## Compilando o código-fonte para o cliente de ElastiCache cluster para PHP

Esta seção aborda como obter e compilar o código-fonte do ElastiCache Cluster Client forPHP.

[Há dois pacotes que você precisa extrair GitHub e compilar; aws-elasticache-cluster-client-libmemcached e - . aws-elasticache-cluster-client memcached-for-php](#)

### Tópicos

- [Compilação da biblioteca libmemcached](#)
- [Compilando o cliente de descoberta automática ElastiCache Memcached para PHP](#)

### Compilação da biblioteca libmemcached

Para compilar a biblioteca aws-elasticache-cluster-client -libmemcached

1. Inicie uma EC2 instância da Amazon.
2. Instale as dependências da biblioteca.

- No Amazon Linux 2015/09 AMI

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- No Ubuntu 14.04 AMI

```
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/aws-labs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz>

## Compilando o cliente de descoberta automática ElastiCache Memcached para PHP

As seções a seguir descrevem como compilar o ElastiCache Memcached Auto Discovery Client.

### Tópicos

- [Compilando o cliente ElastiCache Memcached para 7 PHP](#)
- [Compilando o cliente ElastiCache Memcached para 5 PHP](#)

## Compilando o cliente ElastiCache Memcached para 7 PHP

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 PHP binário para que ela possa ser portada 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
```

## Compilando o cliente ElastiCache Memcached para 5 PHP

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
```

## Configurando clientes ElastiCache

Um ElastiCache cluster é compatível com o protocolo Valkey, OSS Redis e Memcached. O código, os aplicativos e as ferramentas mais populares que você usa atualmente em seu ambiente atual funcionarão perfeitamente com o serviço.

Esta seção discute considerações específicas para se conectar aos nós de cache em ElastiCache

### Tópicos

- [Comandos restritos](#)
- [Localização de endpoints de nó e números de porta](#)
- [Conexão para uso da descoberta automática](#)
- [Conectando-se a nós em um cluster Valkey ou Redis OSS](#)
- [DNSnomes 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 de cache executando o Redis, 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 ElastiCache de gerenciamento da Amazon](#) 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 DNS nomes e números de porta totalmente qualificados estão na seção Endpoint da saída.

### Encontrando terminais de nós de cache e números de portas () ElastiCache API

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 [Identifique automaticamente os nós em seu cluster \(Memcached\)](#).

### Note

Neste momento, a Descoberta automática só está disponível para clusters de cache executando o Memcached.

## Conectando-se a nós em um cluster Valkey ou Redis OSS

### Note

No momento, clusters (API/CLI: grupos de replicação) que oferecem suporte a réplicas de replicação e leitura são compatíveis somente com clusters que executam Valkey ou Redis OSS.

Para clusters, ElastiCache fornece console e API interfaces para obter informações de conexão para nós individuais. CLI

Para atividades somente leitura, os aplicativos podem se conectar a qualquer nó no cluster. No entanto, para atividades de gravação, recomendamos que seus aplicativos se conectem ao endpoint primário (Valkey ou Redis OSS (modo de cluster desativado)) ou ao endpoint de configuração (Valkey ou Redis OSS (modo de cluster ativado)) 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 [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#).

## 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": {}
}
]
}

```

Conectando-se a clusters em um grupo de replicação () ElastiCache API

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>
```

## DNSnomes 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 ElastiCache, o DescribeCacheClusters API (ou o utilitário de linha de describe-cache-clusters comando) retorna uma DNS entrada totalmente qualificada e um número de porta que podem ser usados para a lista de servidores.

### Important

É importante que os aplicativos cliente sejam configurados para resolver com frequência DNS os nomes dos nós de cache ao tentarem se conectar a um endpoint do nó de cache.

## VPCInstalações

ElastiCache garante que o DNS nome e o endereço IP do nó de cache permaneçam os mesmos quando os nós de cache forem recuperados em caso de falha.

## Não VPC instalações

ElastiCache garante que o DNS nome de um nó de cache permaneça inalterado quando os nós de cache forem recuperados em caso de falha; no entanto, o endereço IP subjacente do nó de cache pode mudar.

A maioria das bibliotecas de cliente oferece suporte para conexões persistentes de nó de cache por padrão. Recomendamos usar conexões de nós de cache persistentes ao usar ElastiCache.

O armazenamento em DNS cache do lado do cliente pode ocorrer em vários locais, incluindo bibliotecas de clientes, o tempo de execução da linguagem ou o 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.

## Hierarquização de dados em ElastiCache

ElastiCache com OSS clusters Valkey ou Redis que compõem um grupo de replicação e usam um tipo de nó da família r6gd, têm seus dados em camadas entre a memória e o armazenamento local (unidades de estado sólido). SSD O armazenamento de dados em camadas fornece uma nova opção de preço-desempenho para OSS cargas de trabalho Valkey ou Redis, utilizando unidades de estado sólido (SSDs) de baixo custo em cada nó do cluster, além de armazenar dados na memória. É ideal para cargas de trabalho que acessam regularmente até 20% de seu conjunto de dados geral e para aplicativos que podem tolerar latência adicional ao acessar dados em SSD

Em ElastiCache clusters com hierarquização de dados, ElastiCache monitora o último horário de acesso de cada item que ele armazena. Quando a memória disponível (DRAM) é totalmente consumida, ElastiCache usa um algoritmo (LRU) usado menos recentemente para mover automaticamente itens acessados com pouca frequência da memória para o SSD. Quando os dados ativados SSD são acessados posteriormente, eles ElastiCache são movidos de volta para a memória de forma automática e assíncrona 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 o armazenamento de dados em camadas, as próprias chaves sempre permanecem na memória, enquanto a chave LRU controla o posicionamento dos valores na memória versus no 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, assumindo valores de string de 500 bytes, você pode esperar 300 microssegundos adicionais de latência, em média, para solicitações de dados armazenados em SSD comparação com solicitações de dados na 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). O armazenamento em camadas de dados é compatível com todos os OSS comandos e estruturas de dados do Valkey ou Redis suportados no ElastiCache. Para usar esse recurso, não é necessário promover alterações no lado do cliente.

## Tópicos

- [Práticas recomendadas](#)
- [Limitações](#)
- [Definição de preço](#)
- [Monitorar](#)
- [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:

- O armazenamento de dados em camadas é ideal para cargas de trabalho que acessam regularmente até 20% de seu conjunto geral de dados e para aplicativos que podem tolerar latência adicional ao acessar dados em SSD
- Ao usar a SSD capacidade disponível em nós com camadas de dados, recomendamos que o tamanho do valor seja maior 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 SSD camada.

## 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 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 escalonamento automático é suportado em clusters usando camadas de dados para Valkey versão 7.2 e posterior e Redis OSS versão 7.0.7 e posterior. Para ter mais informações, consulte [Clusters Auto Scaling 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`.
- O salvamento sem bifurcação é compatível com o Valkey versão 7.2 e posterior e o Redis OSS versão 7.0.7 e posterior. Para obter mais informações, consulte [Como a sincronização e o backup são implementados](#).
- Itens maiores que 128 MiB não são movidos para SSD

## Definição de preço

Os nós R6gd têm 4,8x mais capacidade total (memória +SSD) e podem ajudá-lo a obter mais de 60% de economia ao operar com utilização máxima em comparação com os nós R6g (somente memória). Para obter mais informações, consulte [ElastiCache os preços](#).

## Monitorar

ElastiCache oferece métricas projetadas especificamente para monitorar os clusters de desempenho que usam camadas de dados. Para monitorar a proporção de itens em DRAM comparação com SSD, você pode usar a `Currltems` métrica em [Metrics for Valkey e Redis OSS](#). Você pode calcular a porcentagem como:  $(\text{Currltems com Dimensão: Nível} = \text{Memória} * 100) / (\text{Currltems sem filtro de dimensão})$ .

Se a política de despejo configurada permitir, eles ElastiCache começarão a despejar itens quando a porcentagem de itens na memória diminuir abaixo de 5%. Nos nós configurados com a política de não despejo, as operações de gravação receberão um erro de falta de memória.

Ainda é recomendável que você considere a escalabilidade horizontal para clusters habilitados para o Modo Cluster ou a escalabilidade para clusters desativados do Modo Cluster quando a porcentagem de itens na memória diminuir abaixo de 5%. Para obter mais informações sobre

escalabilidade, consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#). Para obter mais informações sobre métricas para OSS clusters Valkey ou Redis que usam camadas de dados, consulte. [Métricas para Valkey e Redis OSS](#)

## Como usar a classificação de dados em níveis

Usando a classificação por níveis de dados usando o AWS Management Console

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 para Valkey ou Redis OSS](#).

Habilitando a hierarquização de dados usando o AWS CLI

Ao criar um grupo de replicação usando o AWS CLI, você usa a classificação por níveis de dados selecionando um tipo de nó da família r6gd, como cache.r6gd.xlarge e definindo 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 restaurar um backup em um novo cluster com o armazenamento de dados em camadas ativado usando o (Console), (AWS CLI) ou (ElastiCache API). 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 AWS Management Console 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 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 de cluster ativado (escala horizontal) — Escolha isso para um cluster Valkey ou Redis OSS (modo de cluster ativado).
  3. Node Type (Tipo de nó) – Especifique cache.r6gd.xlarge ou qualquer outro tipo de nó da família r6gd.
  4. Número de fragmentos — Escolha o número de fragmentos que você deseja 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. Escolha um VPC — Escolha o VPC no qual criar esse cluster.
  10. Grupo de parâmetros — Escolha um grupo de parâmetros que reserve memória suficiente para a OSS sobrecarga do Valkey ou do Redis 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 para 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 o AWS CLI, o armazenamento em camadas de dados é usado por padrão selecionando um tipo de nó da família r6gd, como `cache.r6gd.xlarge` e definindo 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 \  
  --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
  }
}
```

## Preparando um cluster em ElastiCache

A seguir, você encontrará instruções sobre como criar um cluster usando o ElastiCache console, AWS CLI, o ou ElastiCache API o.

Você também pode criar um ElastiCache cluster usando [AWS CloudFormation](#). Para obter mais informações, consulte [AWS:ElastiCache:: CacheCluster](#) no Guia do usuário do AWS 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

- [Determinando seus requisitos ElastiCache de cluster](#)
- [Escolha do tamanho do nó](#)

## Determinando seus requisitos ElastiCache de cluster

### Preparação

Saber as respostas às perguntas a seguir ajuda a facilitar a criação ElastiCache do cluster:

- 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 lançará seu cluster em uma nuvem privada virtual (VPC) baseada na AmazonVPC?

#### Important

Se você for iniciar seu cluster em um VPC, certifique-se de criar um grupo de sub-redes no mesmo VPC antes de começar a criar um cluster. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).

ElastiCache foi projetado para ser acessado de dentro AWS usando a AmazonEC2.

No entanto, se você iniciar em um sistema VPC baseado na Amazon VPC e seu cluster estiver em um VPC, você poderá fornecer acesso externo 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 ElastiCache parâmetros](#).

Se você estiver executando o Valkey ou o RedisOSS, considere configurar `reserved-memory` ou `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciando a memória reservada para Valkey e Redis OSS](#).

- Você precisa criar seu próprio grupo VPC de segurança?

Para obter mais informações, consulte [Segurança em seu VPC](#).

- Como você pretende implementar a tolerância a falhas?

Para obter mais informações, consulte [Atenuar falhas](#).

## Tópicos

- [ElastiCache requisitos de memória e processador](#)
- [Configuração do cluster do Memcached](#)
- [Configuração de cluster Valkey e Redis OSS](#)
- [ElastiCache requisitos de escalabilidade](#)
- [ElastiCache requisitos de acesso](#)

- [Requisitos de região, zona de disponibilidade e zona local para ElastiCache](#)

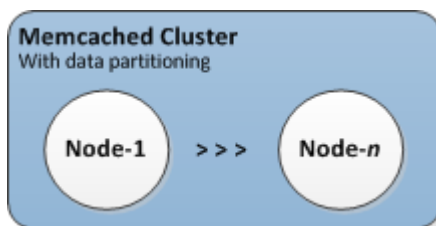
## ElastiCache requisitos de memória e processador

O alicerce básico da 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

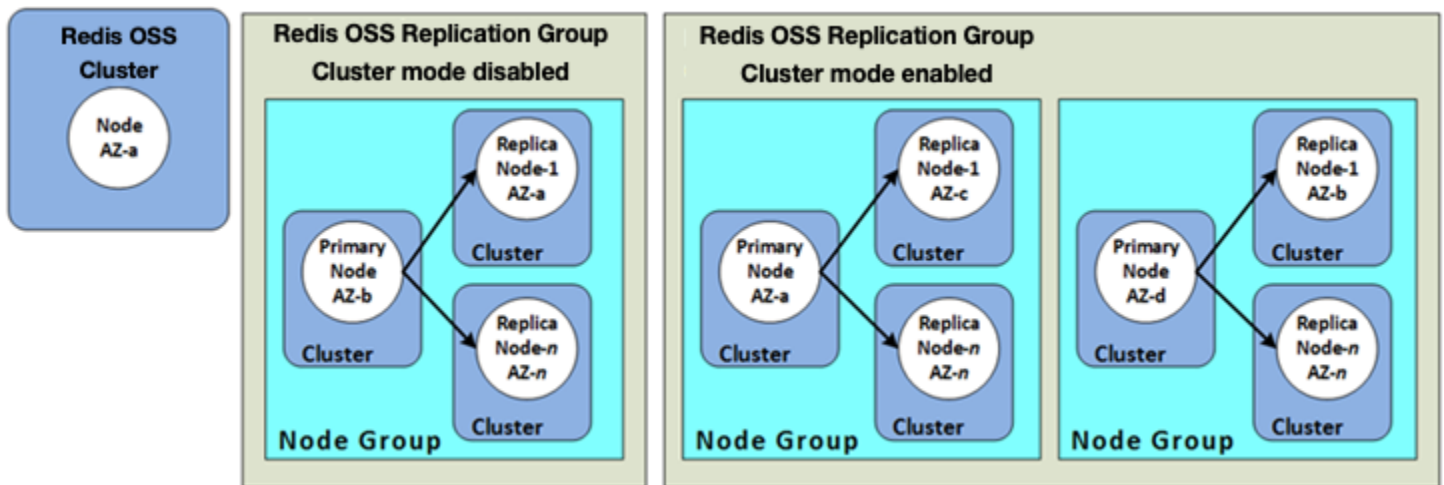
ElastiCache Os clusters (Memcached) são compostos de 1 a 60 nós. Os dados em um cluster Memcached são particionados nos nós do cluster. Seu aplicativo conecta-se a um cluster Memcached usando um endereço de rede chamado de Endpoint. Cada nó em um cluster Memcached tem seu próprio endpoint, que seu aplicativo 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 [Gerenciando clusters em ElastiCache](#).

## Configuração de cluster Valkey e Redis OSS

ElastiCache com Valkey e Redis, os OSS clusters são compostos de 0 a 500 fragmentos (também chamados de grupos de nós). Os dados em um OSS cluster Valkey ou Redis são particionados entre os fragmentos do cluster. Seu aplicativo se conecta a um OSS cluster Valkey ou Redis usando um endereço de rede chamado Endpoint. Os nós em um OSS fragmento Valkey ou Redis cumprem uma das duas funções: um primário de leitura/gravação e todos os outros nós secundários somente para leitura (também chamados de réplicas de leitura). Além dos endpoints do nó, o próprio OSS cluster Valkey ou Redis tem um endpoint chamado endpoint de configuração. Seu aplicativo pode usar esse endpoint para ler ou gravar no cluster, deixando a determinação de qual nó ler ou gravar até ElastiCache (RedisOSS).



Para obter mais informações, consulte [Gerenciando clusters em ElastiCache](#).

### ElastiCache requisitos de escalabilidade

Todos os clusters podem ser expandidos criando um novo cluster com o novo tipo de nó maior. Quando você aumenta a escala de um cluster Memcached, o novo cluster começa vazio. Ao escalar um OSS cluster Valkey ou Redis, você pode implantá-lo a partir de um backup e evitar que o novo cluster comece vazio.

Os clusters do Amazon ElastiCache for Memcached podem ser ampliados ou ampliados. 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.

### ElastiCache requisitos de acesso

Por padrão, os ElastiCache clusters da Amazon são acessados a partir de EC2 instâncias da Amazon. O acesso à rede a um ElastiCache cluster é limitado à conta que criou o cluster. Portanto, antes de acessar um cluster a partir de uma EC2 instância da Amazon, você deve autorizar a EC2 instância da Amazon a acessar o cluster. As etapas para fazer isso variam, dependendo se você iniciou o EC2 - VPC ou o EC2 -Classic.

Se você lançou seu cluster em EC2 - VPC você precisa conceder a entrada de rede ao cluster. Se você lançou seu cluster no EC2 -Classic, precisará conceder ao grupo de segurança Amazon Elastic Compute Cloud associado à instância acesso ao seu grupo de ElastiCache segurança. Para obter instruções detalhadas, consulte [Etapa 3. Autorizar o acesso ao cluster](#) neste guia.

## Requisitos de região, zona de disponibilidade e zona local para ElastiCache

A Amazon ElastiCache oferece suporte a todas AWS as regiões. Ao localizar seus ElastiCache clusters em uma AWS região próxima ao seu aplicativo, você pode 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 as informações a seguir.

- [Escolhendo regiões e zonas de disponibilidade para ElastiCache](#)
- [Usando zonas locais com ElastiCache](#)
- [Atenuar falhas](#)

## Escolha do tamanho do nó

O tamanho do nó que você seleciona para seu ElastiCache cluster afeta os custos, o desempenho e a tolerância a falhas.

### Tamanho do nó (Valkey e RedisOSS)

Para receber informações sobre os benefícios dos processadores Graviton, consulte [Processador Graviton da AWS](#).

Responder às perguntas a seguir pode ajudá-lo a determinar o tipo mínimo de nó necessário para sua implementação do Valkey ou do RedisOSS:

- Você espera workloads limitadas ao throughput com várias conexões de clientes?

Se esse for o caso e você estiver executando a OSS versão 5.0.6 ou superior do Redis, você pode obter melhor taxa de transferência e latência com nosso recurso de E/S aprimorado, quando disponível, usado para descarregar CPUs as conexões do cliente, em nome do mecanismo do Redis. OSS Se você estiver executando a OSS versão 7.0.4 ou superior do Redis, além da E/S aprimorada, você obterá aceleração adicional com a multiplexação de E/S aprimorada, em que cada thread de E/S de rede dedicado envia comandos de vários clientes para o OSS mecanismo do Redis, aproveitando a capacidade do Redis de processar comandos em lotes com eficiência. OSS No ElastiCache (RedisOSS) v7.1 e versões posteriores, ampliamos a funcionalidade aprimorada de threads de E/S para também lidar com a lógica da camada de apresentação. Por camada de apresentação, o que queremos dizer é que os encadeamentos de E/S aprimorados agora não estão apenas lendo a entrada do cliente, mas também analisando



a entrada no formato de comando OSS binário do Redis, que é então encaminhado para o encadeamento 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?

Se esse for o caso e você estiver executando a versão 6.2 ou posterior OSS do mecanismo Redis, você pode aproveitar a hierarquização de dados escolhendo o tipo de nó r6gd. Com o armazenamento de dados em camadas, 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 [Hierarquização de dados em 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 você OSS está executando?

OSSAs versões do Redis 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.

A OSS versão 2.8.22 e posterior do Redis usa um processo de salvamento sem bifurcação que requer menos memória disponível do que o processo anterior.

Para obter mais informações, consulte as informações a seguir.

- [Como a sincronização e o backup são implementados](#)
- [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do 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 a captura de um instantâneo, a sincronização de um cluster primário com uma réplica em um cluster e a ativação do recurso append-only file (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 [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#).

- Sua implementação será um cluster Valkey ou Redis autônomo OSS (modo de cluster desativado) ou um cluster Valkey ou Redis OSS (modo de cluster ativado) com vários fragmentos?

Cluster Valkey ou Redis OSS (modo de cluster desativado)

Se você estiver implementando um cluster Valkey ou Redis OSS (modo de cluster desativado), seu tipo de nó deve ser capaz de acomodar todos os seus dados, além da sobrecarga necessária, conforme descrito no bullet anterior.

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 de cluster ativado) com vários fragmentos

Se você estiver implementando um cluster Valkey ou Redis OSS (modo de cluster ativado) com vários fragmentos, o tipo de nó deverá 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?

[As Zonas Locais](#) permitem que você coloque recursos, como um ElastiCache cluster, em vários locais próximos aos seus 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 M5: cache.m5.large, cache.m5.xlarge, cache.m5.2xlarge, cache.m5.4xlarge, cache.m5.12xlarge, cache.m5.24xlarge

Tipos de nó R5: cache.r5.large, cache.r5.xlarge, cache.r5.2xlarge, cache.r5.4xlarge, cache.r5.12xlarge, cache.r5.24xlarge

Tipos de nó T3: cache.t3.micro, cache.t3.small, cache.t3.medium

Enquanto seu cluster está em execução, você pode monitorar as métricas de uso de memória, utilização do processador, acertos e erros de cache que são publicadas em 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 maiores CPU e de memória.

Ao monitorar o CPU uso, lembre-se de que o Valkey e o Redis OSS são de um único thread. Assim, multiplique o CPU uso relatado pelo número de CPU núcleos para obter esse uso real. Por exemplo, um CPU relatório de quatro núcleos com uma taxa de uso de 20% é, na verdade, o único núcleo que o Redis OSS está funcionando 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 total de memória do seu cluster é calculada multiplicando o número de nós no cluster pela RAM capacidade de cada nó após deduzir a sobrecarga 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, ElastiCache provisiona um substituto para um nó com falha e ele é repovoado. 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:

- 11 nós cache.t2.medium com 3,22 GB de memória e 2 threads cada = 35,42 GB e 22 threads.
- 6 nós cache.m4.large com 6,42 GB de memória e 2 threads cada = 38,52 GB e 12 threads.
- 3 nós cache.r4.large com 12,3 GB de memória e 2 threads cada = 36,90 GB e 6 threads.
- 3 nós cache.m4.xlarge 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	\$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	\$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 de suas opções específicas, consulte os [ElastiCache preços da Amazon](#).

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 facilitar o gerenciamento de endpoints, você pode usar ElastiCache o recurso Auto Discovery, que permite que os programas cliente identifiquem automaticamente todos os nós em um cluster. Para obter mais informações, consulte [Identifique automaticamente os nós em 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, CPU a utilização e a taxa de acerto do cache com ElastiCache as métricas publicadas na Amazon CloudWatch. Para obter mais informações sobre CloudWatch métricas para ElastiCache,

consulte [Monitorando o uso com CloudWatch métricas](#). Para produção e cargas de trabalho maiores, os nós R5 oferecem o melhor desempenho e a melhor relação RAM custo-benefício.

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 seu cluster estiver limitado, CPU mas tiver uma taxa de acerto suficiente, configure um novo cluster com um tipo de nó que forneça mais potência computacional.

## Criação de um cluster para Valkey ou Redis OSS

Os exemplos a seguir mostram como criar um OSS cluster Valkey ou Redis usando o AWS Management Console, e. AWS CLI ElastiCache API

### Criação de um Valkey ou Redis OSS (modo de cluster desativado) (Console)

ElastiCache oferece suporte à replicação quando você usa o mecanismo Valkey ou RedisOSS. Para monitorar a latência entre o momento em que os dados são gravados em um cluster primário de OSS leitura/gravação Valkey ou Redis 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 atual da Hora Universal Universal (UTC). Como um OSS cluster Valkey ou Redis pode ser adicionado a um grupo de replicação posteriormente, essa chave é incluída em todos os OSS clusters Valkey ou Redis, 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 Valkey ou Redis OSS (modo de cluster desativado), siga as etapas em. [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)

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 ter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4. Conecte-se ao nó do 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 Valkey ou Redis OSS (modo de cluster ativado) (console)

Se você estiver executando o Redis OSS 3.2.4 ou posterior, poderá criar um cluster Valkey ou Redis OSS (modo de cluster ativado). Os clusters Valkey ou Redis OSS (modo de cluster ativado) oferecem suporte ao particionamento de seus dados em 1 a 500 fragmentos (API/CLI: grupos de nós), mas com algumas limitações. Para uma comparação entre Valkey ou Redis OSS (modo de

cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado), 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 AWS Management Console 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 Criar VPC e siga as etapas descritas em [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 \(RedisOSS\)](#) com o 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 Valkey e Redis OSS](#) e [Criação de um grupo de ElastiCache parâmetros](#).



**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 [Hierarquização de dados em ElastiCache](#).

- e. Em Número de fragmentos, escolha o número de fragmentos (partições/grupos de nós) que você deseja para esse cluster Valkey ou Redis OSS (modo de cluster ativado).

Para algumas versões do Valkey ou do Redis OSS (modo de cluster ativado), você pode alterar dinamicamente o número de fragmentos em seu cluster:

- Redis OSS 3.2.10 e versões posteriores — Se seu cluster estiver executando o Redis OSS 3.2.10 ou versões posteriores, você poderá alterar dinamicamente o número de fragmentos em seu cluster. Para obter mais informações, consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#).
- Outras OSS versões do Redis — Se seu 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 Valkey ou Redis OSS (modo de cluster ativado).

- 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: semeando um novo cluster autoprojetoado 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-redes, insira o VPCID 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](#).
- [Crie uma sub-rede no seu VPC](#)

Se você for [Usando zonas locais com 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 [Escolhendo regiões e zonas de disponibilidade para 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 AWS KMS chave gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando chaves gerenciadas pelo cliente de 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 superior ou para a versão 6.0 e superior OSS do mecanismo Redis, 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 [Gerenciando grupos de usuários com o console e CLI](#).
- AUTHUsuário padrão — Um mecanismo de autenticação para um servidor Valkey ou RedisOSS. Para obter mais informações, consulte [AUTH](#).
- AUTH— Um mecanismo de autenticação para o servidor Valkey ou RedisOSS. Para obter mais informações, consulte [AUTH](#).

 Note

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

- 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. Você pode usar o grupo de segurança padrão para o seu VPC ou criar um novo.

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

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 janela de manutenção é o horário, geralmente de uma hora de duração, a cada semana em que ElastiCache agenda a manutenção do sistema para seu cluster. Você pode ElastiCache escolher o dia e a hora da janela de manutenção (sem preferência), ou você mesmo pode escolher o dia, a hora e a duração (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 tempos são UCT tempos.

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

9. (Opcional) Para Logs:

- Em Formato de registro, escolha 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 o Multi-AZ
2. Siga as etapas em [Uso do Outposts](#).

Para criar o equivalente usando 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 ter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4. Conecte-se ao nó do 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](#).

## Criando um cluster Valkey ou Redis OSS (modo de cluster desativado) () CLI

Example — Um cluster Valkey ou Redis OSS (modo de cluster desativado) sem réplicas de leitura

O CLI código a seguir cria um cluster de cache Valkey ou Redis OSS (modo de cluster desativado) 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 \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--cache-parameter-group default.redis6.x \  
--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 ^  
--cache-parameter-group default.redis6.x ^  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Criando um cluster Valkey ou Redis OSS (modo de cluster ativado) ()AWS CLI

Clusters Valkey ou Redis OSS (modo de cluster ativado) (API/CLI: grupos de replicação) não podem ser criados usando a operação. `create-cache-cluster` Para criar um cluster Valkey ou Redis OSS (modo de cluster ativado) (API/CLI: grupo de replicação), consulte. [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster ativado\) do zero \(\)AWS CLI](#)

Para obter mais informações, consulte o tópico AWS CLI de ElastiCache referência [create-replication-group](#).

Criando um cluster para Valkey ou Redis OSS () ElastiCache API

Para criar um cluster usando o 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 de cache Valkey ou Redis OSS \(modo de cluster desativado\) \(\) ElastiCache API](#)
- [Criação de um cluster de cache no Valkey ou Redis OSS \(modo de cluster ativado\) \(\) ElastiCache API](#)

Criação de um cluster de cache Valkey ou Redis OSS (modo de cluster desativado) () ElastiCache API

O código a seguir cria um cluster de cache Valkey ou Redis OSS (modo de cluster desativado) (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  
&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 de cache no Valkey ou Redis OSS (modo de cluster ativado) () ElastiCache API

Clusters Valkey ou Redis OSS (modo de cluster ativado) (API/CLI: grupos de replicação) não podem ser criados usando a operação `CreateCacheCluster`. Para criar um cluster Valkey ou Redis OSS (modo de cluster ativado) (API/CLI: grupo de replicação), consulte [Criando um grupo de replicação no Valkey ou no Redis OSS \(modo de cluster ativado\) do zero \(\) ElastiCache API](#)

Para obter mais informações, consulte o tópico ElastiCache API de referência [CreateReplicationGroup](#).

## Criação de um cluster para Memcached

Os exemplos a seguir mostram como criar um cluster usando o AWS Management Console, AWS CLI ElastiCache API 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, siga as etapas em [Criar um cluster](#)

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 ter mais informações, consulte [Etapa 3. Autorizar o acesso ao cluster](#) e [Etapa 4. Conecte-se ao nó do 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 CLI código a seguir cria um cluster de cache 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 () ElastiCache API

Para criar um cluster usando o 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 de cache Memcached \(\) ElastiCache API](#)

## Criação de um cluster de cache Memcached () ElastiCache API

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>
```

## Visualizando os detalhes de um ElastiCache cluster

Você pode visualizar informações detalhadas sobre um ou mais clusters usando o ElastiCache console, AWS CLI, ou ElastiCache API.

### Visualizando os detalhes de um cluster Memcached (console)

Você pode visualizar os detalhes de um cluster Memcached usando o ElastiCache console, o AWS CLI for ElastiCache ou o ElastiCache API

O procedimento a seguir detalha como visualizar os detalhes de um cluster Memcached usando o ElastiCache console.

Para visualizar os detalhes de um cluster Memcached

1. Faça login no AWS Management Console 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 está interessado.
3. No painel do ElastiCache motor, escolha Memcached. Isso exibe uma lista de todos os seus clusters que estão sendo executados 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 ter mais informações, consulte [Monitorando o uso com CloudWatch métricas](#)
7. Escolha a guia Rede e segurança para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-redes do cluster e o grupo VPC de segurança. 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 [Gerenciando a manutenção ElastiCache do cluster](#).
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](#).

## Visualizando detalhes do Valkey ou do Redis OSS (modo de cluster desativado) (console)

Você pode visualizar os detalhes de um cluster Valkey ou Redis OSS (modo de cluster desativado) usando o ElastiCache console, o AWS CLI for ElastiCache ou o ElastiCache API

O procedimento a seguir detalha como visualizar os detalhes de um cluster Valkey ou Redis OSS (modo de cluster desativado) usando o ElastiCache console.

Para ver os detalhes de um cluster Valkey ou Redis OSS (modo de cluster desativado)

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache mecanismo, escolha Valkey ou Redis OSS para exibir uma lista de todos os seus clusters que estão sendo executados nesse 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 executando o OSS mecanismo Valkey ou Redis, 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 ter 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 [Gerenciando a manutenção ElastiCache do cluster](#).

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](#).

Visualizando detalhes de um cluster Valkey ou Redis OSS (modo de cluster ativado) (console)

Você pode visualizar os detalhes de um cluster Valkey ou Redis OSS (modo de cluster ativado) usando o ElastiCache console, o AWS CLI for ElastiCache ou o ElastiCache API

O procedimento a seguir detalha como visualizar os detalhes de um cluster Valkey ou Redis OSS (modo de cluster ativado) usando o ElastiCache console.

Para ver os detalhes de um cluster Valkey ou Redis OSS (modo de cluster ativado)

1. Faça login no AWS Management Console 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 está interessado.
3. No painel do ElastiCache mecanismo, escolha Valkey ou Redis OSS para exibir uma lista de todos os seus clusters que estão sendo executados nesse mecanismo.
4. Para ver detalhes de um cluster Valkey ou Redis OSS (modo de cluster ativado), escolha a caixa à esquerda do nome do cluster. Certifique-se de escolher um cluster 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 ter 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 Rede e segurança para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-rede do cluster, o grupo de VPC segurança e qual método de criptografia, se houver, está habilitado no cluster. Para ter mais informações, consulte [Sub-redes e grupos de sub-redes](#) e [Segurança de dados na Amazon ElastiCache](#).
10. Escolha a guia Maintenance (Manutenção) para ver detalhes sobre as configurações de manutenção do cluster. Para obter mais informações, consulte [Gerenciando a manutenção ElastiCache do cluster](#).
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](#).

Visualizando os detalhes de um ElastiCache cluster (AWS CLI)

O código a seguir lista os detalhes de *my-cluster*:

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

Substituir *my-cluster* com o nome do seu cluster em um caso em que o cluster é criado com 1 nó de cache e 0 fragmentos usando o `create-cache-cluster` comando.

```
{
  "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 é criado usando o AWS Management Console (nó do cluster ativado ou desativado com 1 ou mais fragmentos), use o comando a seguir para descrever os detalhes do cluster (substituir *my-cluster* com o nome do grupo de replicação (nome do seu cluster)):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Para obter mais informações, consulte o ElastiCache tópico AWS CLI for [describe-cache-clusters](#).

Visualizando os detalhes de um ElastiCache cluster (ElastiCache API)

Você pode ver os detalhes de um cluster usando a ElastiCache API DescribeCacheClusters ação. 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 ElastiCache API de referência [DescribeCacheClusters](#).

## Modificando um cluster ElastiCache

Além de adicionar ou remover nós de um ElastiCache cluster, pode haver momentos em que você precise 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 determinada alteração de parâmetro é aplicada, consulte a seção Alterações entram em vigor da coluna Detalhes nas tabelas de [Parâmetros específicos do Memcached](#), [Parâmetros Valkey](#) e [Redis OSS](#). Para obter informações sobre como reinicializar um cluster, consulte [Reinicializando nós](#).

### Usando o ElastiCache AWS Management Console

#### Como modificar um cluster

1. Faça login no AWS Management Console 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. 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 OSS clientes Valkey ou Redis se conectem usando o modo de cluster ativado e o modo de cluster desativado. Depois de migrar todos os OSS clientes Valkey ou Redis para usar o modo de cluster ativado, você pode concluir a configuração do modo de cluster e definir o modo de cluster como Ativado.

- 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.


- VPCGrupo (s) de segurança
- 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
- Período de retenção de backup
- Janela de backup
- Tópico para SNS notificação

- Compatibilidade de versões do Memcached Engine
- Tipo de rede

 Note

Se você estiver migrando de IPv4 para IPv6, deverá selecionar ou criar grupos de sub-redes compatíveis com o IPv6. Para obter mais informações, consulte [Escolhendo um tipo de rede em ElastiCache](#).

- VPC Grupo (s) de segurança
- Grupo do parâmetro
- Janela de manutenção
- Tópico para SNS notificação

A caixa Apply Immediately (Aplicar imediatamente) aplica-se apenas a modificações feitas 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 ativar/desativar a entrega de registros 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 de detalhes do cluster, escolha a guia Registros.
3. Para ativar ou desativar registros lentos, escolha Ativar ou Desativar.

Se você escolher habilitar:

- a. Em Formato de registro, escolha um JSON ou Texto.
- b. Em Tipo de destino do registro, escolha CloudWatch Logs ou Kinesis Firehose.
- c. Em Destino do registro, você pode escolher Criar novo e inserir o nome do seu grupo de CloudWatch Logs registros ou o nome do stream do Kinesis Data Firehose. Você também

pode escolher **Selecionar existente** e, em seguida, escolher o nome do seu grupo de CloudWatchLogs registros ou o nome do stream do Kinesis Data Firehose.


d. Escolha **Habilitar**.

Para alterar sua configuração do Redis:

1. Escolha **Modificar**.
2. Em **Formato de registro**, escolha um **JSON** ou **Texto**.
3. Em **Tipo de destino**, escolha **CloudWatch Logs** ou **Kinesis Firehose**.
4. Em **Destino do registro**, escolha **Criar novo** e insira o nome do seu grupo de CloudWatchLogs registros ou o nome do stream do Kinesis Data Firehose. Ou escolha **Selecionar existente** e, em seguida, escolha o nome do seu grupo de CloudWatchLogs registros ou o nome do stream do Kinesis Data Firehose.

Usando o AWS CLI com ElastiCache

Você pode modificar um cluster existente usando a AWS CLI `modify-cache-cluster` operação. 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 fazer o upgrade 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 fazer o upgrade para versões mais recentes do motor Valkey ou RedisOSS. Se você atualizar as versões principais do mecanismo, por exemplo, do Redis OSS 5.0.6 para o Redis OSS 6.0, precisará 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 o ElastiCache tópico AWS CLI for [modify-cache-cluster](#).

Usando o ElastiCache API

Você pode modificar um cluster existente usando a ElastiCache API `ModifyCacheCluster` operação. 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 fazer o upgrade 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 fazer o upgrade para versões mais recentes do motor Valkey ou RedisOSS. Se você atualizar as versões principais do mecanismo, por exemplo, do Redis OSS 5.0.6 para o Redis OSS 6.0, precisará 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 ElastiCache API de referência [ModifyCacheCluster](#).



## Adicionando nós a um ElastiCache cluster

Adicionar nós a um cluster do Memcached aumenta o número de partições dele. 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 seu cluster Valkey ou Redis OSS (modo de cluster ativado), consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)

Você pode usar o ElastiCache Management Console AWS CLI ou ElastiCache API para adicionar nós ao seu cluster.

### Usando o ElastiCache AWS Management Console

Se você quiser adicionar um nó a um cluster Valkey ou Redis de nó único (modo de cluster desativado) OSS (um sem replicação ativada), é um processo de duas etapas: primeiro adicionar a replicação e, em seguida, adicionar um nó de réplica.

### Tópicos

- [Para adicionar replicação a um OSS cluster Valkey ou Redis sem fragmentos](#)
- [Para adicionar nós a um ElastiCache cluster \(console\)](#)

O procedimento a seguir adiciona a replicação a um Valkey ou Redis de nó único OSS que não tem a replicação ativada. 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 OSS cluster Valkey ou Redis sem fragmentos

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

Uma lista dos clusters que executam esse mecanismo 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 se aplica a um OSS cluster Redis que não tem a replicação habilitada:

- Ele está executando o RedisOSS, 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 AWS Management Console 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 seu cluster for um cluster Valkey ou Redis OSS (modo de cluster ativado), consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)

Se seu cluster for um cluster Valkey ou Redis OSS (modo de cluster desativado) com zero fragmentos, primeiro conclua as etapas em [Para adicionar replicação a um OSS cluster Valkey ou Redis 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	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>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.</p>
Cenário 2	Delete	Criar	<p>A nova solicitação de criação, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>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.</p>
Cenário 3	Criar	Delete	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de criação pendente.</p> <p>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.</p>

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 4	Criar	Criar	<p>A nova solicitação de criação é adicionada à solicitação de criação pendente.</p> <p>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.</p> <div data-bbox="764 667 1507 1220" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b></p> <p>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.</p> </div>

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 de cache 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
```

Essa operação produz uma saída semelhante à seguinte (JSONformato):

```
{  
  "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 você quiser adicionar nós a um cluster Valkey ou Redis existente OSS (modo de cluster desativado) que não tenha a replicação ativada, primeiro crie o grupo de replicação especificando o cluster existente como primário. Para obter mais informações, consulte [Criação de um grupo de replicação usando um cluster de OSS cache Valkey ou Redis 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
```

Essa operação produz uma saída semelhante à seguinte (JSONformato):

```
{  
  "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 o ElastiCache API

Se você quiser adicionar nós a um cluster Valkey ou Redis existente OSS (modo de cluster desativado) que não tenha a replicação ativada, primeiro crie o grupo de replicação especificando o cluster existente como Primário. Para obter mais informações, consulte [Adicionar réplicas a um cluster autônomo Valkey ou Redis OSS \(modo de cluster desativado\) \(\) ElastiCache API](#). 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 IncreaseReplicaCount API operação 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 menor que o número atual de nós, use o DecreaseReplicaCount API 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.
  - RegionEspecifica 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 ElastiCache API tópico [IncreaseReplicaCount](#).

## Usando o ElastiCache API

Para adicionar nós a um cluster (ElastiCache API)

- Chame a `ModifyCacheCluster` API operação 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 o 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 ElastiCache API tópico [ModifyCacheCluster](#).

## Removendo nós de um ElastiCache cluster

Você pode excluir um nó de um cluster Valkey, Redis OSS ou Memcached usando o AWS Management Console, o AWS CLI ou o ElastiCache API

### 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 AWS Management Console

Para remover nós de um cluster (console)

1. Faça login no AWS Management Console 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 ElastiCache console, você só pode excluir um nó por vez, então escolher vários nós significa que você não pode 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 o RedisOSS, se você estiver excluindo os resultados do nó no cluster que não são mais compatíveis com Multi-AZ, primeiro desmarque a caixa de seleção Multi-AZ e depois exclua 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	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>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.</p>
Cenário 2	Delete	Criar	<p>A nova solicitação de criação, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>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.</p>
Cenário 3	Criar	Delete	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de criação pendente.</p> <p>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.</p>

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 4	Criar	Criar	<p>A nova solicitação de criação é adicionada à solicitação de criação pendente.</p> <p>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.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b></p> <p>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.</p> </div>

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 [Visualizando os detalhes de um ElastiCache cluster](#).
2. Use a `decrease-replica-count` CLI operação 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
```

Essa operação produz uma saída semelhante à seguinte (JSONformato):

```
{  
  "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 o ElastiCache API com Valkey ou Redis OSS

Para remover nós usando o ElastiCache API, chame a `DecreaseReplicaCount` API operação 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 ElastiCache API tópico [DecreaseReplicaCount](#).

## Usando o ElastiCache API com o Memcached

Para remover nós usando o ElastiCache API, chame a `ModifyCacheCluster` API operação 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 ElastiCache API tópico [ModifyCacheCluster](#).

## Cancelamento de operações pendentes de adição ou exclusão de nós no ElastiCache

Se você optou por não aplicar uma alteração de ElastiCache cluster imediatamente, a operação tem status pendente até ser executada na próxima janela de manutenção. Você pode cancelar qualquer operação pendente.

Para cancelar uma operação pendente

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a AWS região na qual você deseja cancelar uma operação pendente de adição ou exclusão de nó.
3. No painel de navegação, escolha o mecanismo em execução no cluster que tem operações pendentes que você deseja cancelar. É exibida uma lista de clusters que executam o mecanismo escolhido.
4. Na lista de clusters, escolha o nome do cluster (não a caixa à esquerda do nome do cluster) com operações pendentes que você gostaria de cancelar.
5. 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.
6. Escolha a guia Nós.
7. Para cancelar todas as operações pendentes, clique em Cancel Pending. A caixa de diálogo Cancel Pending é exibida.
8. Confirme que deseja cancelar todas as operações pendentes, escolhendo o botão Cancel Pending. Ou, para manter as operações, escolha Cancel.

## 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 AWS Management Console

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 AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel do ElastiCache mecanismo, escolha o mecanismo em que o cluster que você deseja excluir está em execução.

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 cluster `my-cluster` de ElastiCache cache.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

A `delete-cache-cluster` CLI ação exclui somente um cluster de cache. Para excluir vários clusters de cache, chame `delete-cache-cluster` para cada cluster de cache que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster de cache 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 o 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 `DeleteCacheCluster` API operação exclui somente um cluster de cache. Para excluir vários clusters de cache, chame `DeleteCacheCluster` para cada cluster de cache que você deseja excluir. Você não precisa esperar a finalização da exclusão de um cluster de cache antes de excluir outro.

Para obter mais informações, consulte o tópico ElastiCache API de referência [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 (AmazonVPC), você pode acessar sua ElastiCache instância a partir de uma EC2 instância da Amazon na mesma AmazonVPC. Ou, usando o VPC peering, você pode acessar sua ElastiCache instância de uma Amazon EC2 em outra AmazonVPC.

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 (AmazonVPC), você pode se conectar ao seu ElastiCache cluster somente a partir de uma EC2 instância da Amazon que esteja sendo executada na mesma AmazonVPC. 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, você VPC é estendido para essa zona local e você VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade, e os gateways, tabelas de rotas e outras considerações de grupos de segurança relevantes serão ajustados automaticamente.

Para conceder a entrada de rede de um grupo de VPC segurança da Amazon para um cluster

1. Faça login no AWS Management Console 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 grupo de segurança para sua AmazonVPC. 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. Selecione a opção Editar.
  - b. Escolha Adicionar regra.
  - c. Na coluna Tipo, escolha TCPRegra 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 para Memcached é **11211** A porta padrão para Valkey e Redis é. OSS **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 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 do. 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 VPC de Segurança que será usado exclusivamente por ElastiCache. Para obter mais informações, consulte [Grupos de segurança personalizados](#).

- f. Escolha Salvar.

Quando você executa uma EC2 instância da Amazon em sua AmazonVPC, 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 um 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 de Network Address Translation (NAT). Não existem exceções para esse requerimento.
- A NAT instância deve ser executada da VPC mesma forma que o cluster.
- A NAT instância deve ser executada em uma sub-rede pública separada do cluster.
- Um endereço IP elástico (EIP) deve estar associado à NAT instância. O recurso de encaminhamento de portas do iptables é usado para encaminhar uma porta na NAT instância para a porta do nó de cache dentro do VPC

### Considerações

As considerações a seguir devem ser lembradas ao acessar seus ElastiCache recursos de fora ElastiCache.

- Os clientes se conectam à EIP porta de cache da NAT instância. O encaminhamento de portas na NAT instância encaminha o tráfego para o nó de cluster de cache 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 NAT instância está atuando como um proxy entre clientes e vários clusters. A adição de um proxy afeta o desempenho do cluster de cache. O impacto aumenta com o número de clusters de cache que você está acessando por meio da NAT instância.

- O tráfego dos clientes para a NAT instância não é criptografado. Portanto, você deve evitar o envio de dados confidenciais por meio da NAT instância.
- A NAT instância adiciona a sobrecarga de manter outra instância.
- A NAT instância serve como um único ponto de falha. Para obter informações sobre como configurar a alta disponibilidade NAT emVPC, consulte [Alta disponibilidade para VPC NAT instâncias da Amazon: um exemplo](#).

## Como acessar ElastiCache recursos de fora AWS

O procedimento a seguir demonstra como se conectar aos seus ElastiCache recursos usando uma NAT instância.

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 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 OSS cluster Valkey ou Redis com:
  - Endereço IP: 10.0.1.230
  - Porta padrão — 6379
  - Grupo de segurança: sg-bd56b7da

- AWS endereço IP da instância — sg-bd56b7da
- Seu cliente confiável possui o endereço IP 198.51.100.27.
- Sua NAT instância tem o endereço IP elástico 203.0.113.73.
- Sua NAT instância tem o grupo de segurança sg-ce56b7a9.

Para se conectar aos seus ElastiCache recursos usando uma NAT instância

1. Crie uma NAT instância da VPC mesma forma que seu cluster de cache, mas em uma sub-rede pública.

Por padrão, o VPC assistente iniciará um tipo de nó cache.m1.small. Você deve selecionar um tamanho de nó conforme as suas necessidades. Você deve usar EC2 NAT AMI para poder acessar ElastiCache de fora AWS.

Para obter informações sobre como criar uma NAT instância, consulte [NATInstâncias](#) no Guia AWS VPC do usuário.

2. Crie regras de grupo de segurança para o cluster e a NAT instância de cache.

O grupo de segurança da NAT instância e a instância do cluster devem ter as seguintes regras:

- Duas regras de entrada
  - Com o Memcached, a primeira regra é permitir TCP conexões de clientes confiáveis para cada porta de cache encaminhada da NAT instância (11211 - 11213).
  - Com o Valkey e o RedisOSS, a primeira regra é permitir TCP conexões de clientes confiáveis para cada porta de cache encaminhada da NAT instância (6379 - 6381).
  - Uma segunda regra para permitir o SSH acesso a clientes confiáveis.

NATgrupo de segurança de instância - regras de entrada com Memcached

Tipo	Protocolo	Intervalo de portas	Origem
TCPRegra personalizada	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

## NATgrupo de segurança de instâncias — regras de entrada com Valkey ou Redis OSS

Tipo	Protocolo	Intervalo de portas	Origem
TCPRegra personalizada	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 TCP conexões com a porta de cache (11211).

## NATgrupo de segurança da instância - regra de saída

Tipo	Protocolo	Intervalo de portas	Destination (Destino)
TCPRegra personalizada	TCP	11211	sg-ce56b7a9 (grupo de segurança da instância do cluster)

- Com Valkey ou RedisOSS, uma regra de saída para permitir TCP conexões com a porta de cache (6379).

## NATgrupo de segurança da instância - regra de saída

Tipo	Protocolo	Intervalo de portas	Destination (Destino)
TCPRegra personalizada	TCP	6379	sg-ce56b7a9 (grupo de segurança da instância do cluster)

- Com o Memcached, uma regra de entrada para o grupo de segurança do cluster que permite TCP conexões da NAT instância à porta de cache (11211).

## Grupo de segurança da instância do cluster: regra de entrada

Tipo	Protocolo	Intervalo de portas	Origem
TCPRegra personalizada	TCP	11211	sg-bd56b7da (Grupo de segurança) NAT

- Com Valkey ou RedisOSS, uma regra de entrada para o grupo de segurança do cluster que permite TCP conexões da NAT instância à porta de cache (6379).

## Grupo de segurança da instância do cluster: regra de entrada

Tipo	Protocolo	Intervalo de portas	Origem
TCPRegra personalizada	TCP	6379	sg-bd56b7da (grupo de segurança do cluster)

## 3. Valide as regras.

- Confirme se o cliente confiável é capaz de SSH acessar a NAT instância.
- Confirme se o cliente confiável é capaz de se conectar ao cluster a partir da NAT instância.

## 4. Memcached

Adicione uma regra iptables à NAT instância.

Uma regra iptables deve ser adicionada à NAT tabela para cada nó no cluster para encaminhar a porta de cache da NAT instância para o 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 NAT instância e à porta do cluster correspondente ao nó do cluster apropriado. Por exemplo, a cadeia de conexão para PHP pode ter a seguinte aparência:

```
$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 NAT instância.

Uma regra iptables deve ser adicionada à NAT tabela para cada nó no cluster para encaminhar a porta de cache da NAT instância para o 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 OSS cluster Redis 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 NAT instância e à porta do cluster correspondente ao nó do cluster apropriado. Por exemplo, a cadeia de conexão para PHP pode ter a seguinte aparência:

```
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 da

Os tópicos a seguir podem ser de seu interesse.

- [Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC](#)
- [Acessando um ElastiCache cache a partir de um aplicativo em execução no data center do cliente](#)
- [NATInstâncias](#)
- [Configurando clientes ElastiCache](#)
- [Alta disponibilidade para VPC NAT instâncias da Amazon: um exemplo](#)

## Encontrando pontos de extremidade de conexão em ElastiCache

Seu aplicativo se conecta ao seu ElastiCache cluster usando endpoints. Um endpoint é o endereço exclusivo de um nó ou cluster.

Quais endpoints usar com Valkey ou Redis OSS

- nó autônomo, use o endpoint do nó para operações de leitura e gravação.
- Clusters Valkey, Valkey ou Redis OSS (modo de cluster desativado) usam 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 pontos finais individuais do Node para operações de leitura (noAPI//, CLI eles são chamados de pontos finais de leitura).
- Clusters Valkey ou Redis OSS (habilitado para o modo de cluster) usam o Endpoint de configuração do cluster para todas as operações que oferecem suporte a comandos habilitados para o modo de cluster. Você deve usar um cliente que ofereça suporte ao Valkey Cluster ou ao Redis OSS Cluster (Redis 3.2OSS). Você ainda pode ler a partir de pontos de extremidade de nós individuais (no API /, CLI eles são chamados de pontos de extremidade de leitura).

As seções a seguir orientam pelo processo de descoberta dos endpoints necessários para o mecanismo que você está executando.

Quais endpoints usar com o Memcached

Para o cache ElastiCache sem servidor com o Memcached, basta adquirir o endpoint DNS e a porta do cluster no console.

A partir do AWS CLI, use o `describe-serverless-caches` comando 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
```

A saída da operação acima deve ser semelhante a esta (JSONformato):

```
{
  "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.

## Encontrando os endpoints de um cluster Valkey ou Redis OSS (modo de cluster desativado) (console)

Se um cluster Valkery, Valkey ou Redis OSS (modo de cluster desativado) tiver somente um nó, o endpoint do nó será usado para leituras e gravações. Se um cluster Valkery, Valkey ou Redis OSS (modo de cluster desativado) tiver vários nós, haverá três tipos de endpoints: o endpoint primário, o endpoint do leitor e os endpoints do nó.

O endpoint primário é um DNS nome 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 (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 as várias réplicas de leitura do seu cluster ElastiCache (RedisOSS) 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 DNS registro que será resolvido para um endereço IP de um dos nós de réplica de forma redonda.

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 encontrar os endpoints de um cluster Valkery, Valkey ou Redis OSS (modo de cluster desativado)

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

A tela de clusters aparecerá com uma lista de clusters Valkery, Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado).

- 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 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 [lock icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [lock icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Endpoints primários e Reader para um cluster Valkery, Valkey ou Redis OSS (modo de cluster desativado)

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

- Se o cluster Valkery, Valkey ou Redis OSS (modo de cluster desativado) tiver nós de réplica, você poderá encontrar os endpoints do nó de réplica do cluster escolhendo o nome do cluster e, em seguida, escolhendo a guia Nodes.

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

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	[redacted].amazonaws.com
<input type="checkbox"/>	test-no-002	available	replica	6379	[redacted].amazonaws.com
<input type="checkbox"/>	test-no-003	available	replica	6379	[redacted].amazonaws.com

Endpoints de nós para um cluster Valkery, Valkey ou Redis OSS (modo de cluster desativado)

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 [Conectando-se aos nós do Memcached](#).

Um endpoint primário Valkery, Valkey ou Redis OSS (modo de cluster desativado) se parece com o 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
```

Encontrando endpoints para um cluster Valkey ou Redis OSS (modo de cluster ativado) (console)

Um cluster Valkey ou Redis OSS (modo de cluster ativado) 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 encontrar o endpoint de um cluster Valkey ou Redis OSS (modo de cluster ativado)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters 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.

### Encontrando os 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 [Conectando-se aos nós do Memcached](#).

Endpoints de configuração e de nó são muito 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 seu endpoint de configuração do Memcached, para que seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir no. .cfg. CNAME

## Localizar endpoints (AWS CLI)

Para o Memcached, você pode usar o for AWS CLI Amazon ElastiCache para descobrir os endpoints para nós e clusters.

Para o Redis, você pode usar o AWS CLI for Amazon ElastiCache para descobrir os endpoints para nós, clusters e também grupos de replicação.

### Tópicos

- [Encontrar endpoints para nós e clusters \(AWS CLI\)](#)
- [Encontrando os endpoints para grupos de OSS replicação Valkey ou Redis \(\)AWS CLI](#)

### Encontrar endpoints para nós e clusters (AWS CLI)

Você pode usar o AWS CLI para descobrir os endpoints de um cluster e seus nós com o `describe-cache-clusters` comando. Para OSS clusters Valkey ou Redis, 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
```

A saída da operação acima deve ser semelhante a esta (JSONformato).

```
{
```



```
"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 seu endpoint de configuração do Memcached, para que seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir no. `.cfg`. CNAME Por exemplo, `mycluster.cfg.local` no seu arquivo `php.ini` para o parâmetro `session.save_path`.

### Example

Para Valkey e RedisOSS, o comando a seguir recupera as informações do cluster de nó único `mycluster`.

### Important

O parâmetro `--cache-cluster-id` pode ser usado com ID de cluster Valkey ou Redis de nó único OSS (modo de cluster desativado) 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, `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
```

A saída da operação acima deve ser semelhante a esta (JSONformato).

```
{  
  "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](#).

Encontrando os endpoints para grupos de OSS replicação Valkey ou Redis ()AWS CLI

Você pode usar o AWS CLI para descobrir os endpoints de um grupo de replicação e seus clusters com o `describe-replication-groups` comando. 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
```

A saída dessa operação deve ser semelhante a esta (JSONformato).

```
{  
  "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 AWS CLI Comandos.

## Encontrando endpoints () ElastiCache API

Para o Memcached, você pode usar ElastiCache API a Amazon para descobrir os endpoints para nós e clusters.

Para o Redis, você pode usar a Amazon ElastiCache API para descobrir os endpoints para nós, clusters e também grupos de replicação.

### Tópicos

- [Encontrando endpoints para nós e clusters \(\) ElastiCache API](#)
- [Encontrando endpoints para grupos de OSS replicação Valkey ou Redis \(\) ElastiCache API](#)

## Encontrando endpoints para nós e clusters () ElastiCache API

Você pode usar o ElastiCache API para descobrir os endpoints de um cluster e seus nós com a `DescribeCacheClusters` ação. Para OSS clusters Valkey ou Redis, 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 o endpoint de configuração (*ConfigurationEndpoint*) e os endpoints de nós individuais () do cluster do Memcached *Endpoint 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 seu endpoint de configuração do Memcached, para que seu cliente de descoberta automática reconheça o CNAME como um endpoint de

configuração, você deverá incluir no. `.cfg`. CNAME Por exemplo, `mycluster.cfg.local` no seu arquivo `php.ini` para o parâmetro `session.save_path`.

## Encontrando endpoints para grupos de OSS replicação Valkey ou Redis () ElastiCache API

Você pode usar o ElastiCache API para descobrir os endpoints de um grupo de replicação e seus clusters com a `DescribeReplicationGroups` ação. 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 endpoint do leitor (`ReaderEndpoint`) e os endpoints individuais do nó (`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](#).

## Trabalhando com fragmentos em ElastiCache

Um fragmento (API/CLI: grupo de nós) é uma coleção de um a seis ElastiCache com nós Valkey ou OSS Redis. Um cluster Valkey ou Redis OSS (modo de cluster desativado) nunca terá mais de um fragmento. Com os fragmentos, você pode separar 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 ou Redis 5.0.6 ou superiorOSS. 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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

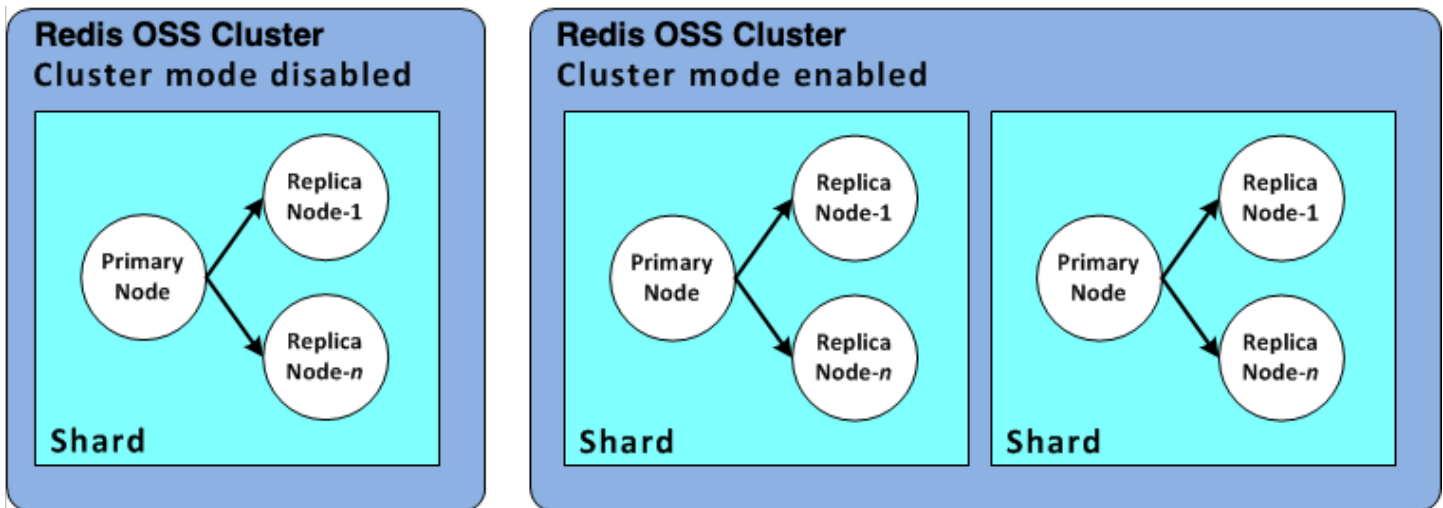
Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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 Valkey ou Redis OSS (modo de cluster ativado) usando o ElastiCache console, você especifica o número de fragmentos no cluster e o número de nós nos fragmentos. Para obter mais informações, consulte [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#). Se você usar o ElastiCache API ou AWS CLI para criar um cluster (chamado de grupo de replicação noAPI/CLI), poderá configurar o número de nós em um fragmento (API/CLI: grupo de nós) de forma independente. Para obter mais informações, consulte as informações a seguir.

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Cada nó em um fragmento tem as mesmas especificações de computação, armazenamento e memória. O ElastiCache API permite controlar atributos de todo o fragmento, como o número de nós, configurações de segurança e janelas de manutenção do sistema.



Configurações de fragmentos do Valkey ou do Redis OSS

Para ter mais informações, consulte [Refragmentação offline para Valkey ou Redis OSS \(modo de cluster ativado\)](#) e [Refragmentação online para Valkey ou Redis OSS \(modo de cluster ativado\)](#).

## Localização de um ID de fragmento

Você pode encontrar o ID de um fragmento usando o AWS Management Console, o AWS CLI ou o ElastiCache API

Usando o AWS Management Console

## Tópicos

- [Para Valkey ou Redis OSS \(modo de cluster desativado\)](#)
- [Para Valkey ou Redis OSS \(modo de cluster ativado\)](#)

Para Valkey ou Redis OSS (modo de cluster desativado)

Os fragmentos do grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) estão sempre. IDs 0001

Para Valkey ou Redis OSS (modo de cluster ativado)

O procedimento a seguir usa o AWS Management Console para encontrar o ID do fragmento do grupo de replicação do Valkey ou Redis OSS (modo de cluster ativado).

Para encontrar o ID do fragmento em um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis e, em seguida OSS, escolha o nome do grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) para o qual você deseja encontrar o fragmento. IDs
3. Na coluna Shard Name (Nome do fragmento), o ID do fragmento tem os últimos quatro dígitos do nome do fragmento.

### Usando o AWS CLI

Para encontrar IDs de fragmento (grupo de nós) para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado), use a AWS CLI operação `describe-replication-groups` com o seguinte parâmetro opcional.

- **--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 fragmento (grupo de nós) são *highlighted* aqui para facilitar sua localização.

```
{
```

```
"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": {}
  }
]
}
```

## Usando o ElastiCache API

Para encontrar IDs de fragmento (grupo de nós) para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado), use a AWS CLI operação `describe-replication-groups` com o seguinte parâmetro opcional.

- **ReplicationGroupId**: um parâmetro opcional que, quando usado, limita a saída aos detalhes do grupo de replicação especificado. Se esse parâmetro for 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>
```

## Comparando os caches autoprojetados do Valkey, do Redis OSS e do Memcached

A Amazon ElastiCache oferece suporte aos mecanismos de cache Valkey, Redis OSS e Memcached. 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 cache, um cluster autoprojetado ou um grupo de replicação, você pode fazer o upgrade para uma versão mais recente do mecanismo, mas não pode fazer o downgrade para uma versão mais antiga do mecanismo. Se quiser usar uma versão mais antiga do mecanismo, exclua o cache existente, o cluster autoprojetado ou o grupo de replicação e crie-o 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 com Valkey 7.2 ou Redis OSS versão 7.0 (aprimorado)

[Você quer usar Functions, Sharded Pub/Sub ou melhorias. ACL](#) Para obter mais informações, consulte [Redis OSS versão 7.0 \(aprimorada\)](#).

- ElastiCache (RedisOSS) versão 6.2 (aprimorada)

Você quer a capacidade de hierarquizar os dados entre a memória e SSD usar o tipo de nó r6gd. Para mais informações, consulte [Classificação de dados em níveis](#).

- ElastiCache (RedisOSS) versão 6.0 (aprimorado)

Você deseja autenticar usuários com o controle de acesso baseado em função.

Para obter mais informações, consulte [Redis OSS versão 6.0 \(aprimorada\)](#).

- ElastiCache (RedisOSS) versão 5.0.0 (aprimorado)

Você quer usar os [OSSfluxos do Redis](#), uma estrutura de dados de log que permite aos produtores acrescentar novos itens em tempo real e também permite que os consumidores consumam mensagens de forma bloqueadora ou não bloqueadora.

Para obter mais informações, consulte [Redis OSS versão 5.0.0 \(aprimorada\)](#).

- ElastiCache (RedisOSS) versão 4.0.10 (aprimorado)


Oferece suporte à criptografia e à adição ou remoção dinâmica de fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado).

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 (RedisOSS) versão 3.2.10 (aprimorada)

Oferece suporte à capacidade de adicionar ou remover dinamicamente fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado).

 Important

Atualmente ElastiCache (RedisOSS) 3.2.10 não oferece suporte à criptografia.

Para obter mais informações, consulte as informações a seguir.

- [OSSVersão 3.2.10 do Redis \(aprimorada\)](#)
- Melhores práticas de refragmentação on-line para RedisOSS. Para obter mais informações, consulte o seguinte:

- [Práticas recomendadas: reestilhecimento online](#)
  - [Refragmentação online e rebalanceamento de fragmentos para Valkey ou Redis \(modo de cluster ativado\) OSS](#)
  - [Para obter mais informações sobre como escalar OSS clusters do Redis, consulte Dimensionamento.](#)
- ElastiCache (RedisOSS) versão 3.2.6 (aprimorada)

Se você precisar da funcionalidade das OSS versões anteriores do Redis, além dos seguintes recursos, escolha ElastiCache (RedisOSS) 3.2.6:

- Criptografia em trânsito. Para obter mais informações, consulte [Amazon ElastiCache \(RedisOSS\) In-Transit Encryption](#).
  - Criptografia em repouso. Para obter mais informações, consulte [Amazon ElastiCache \(RedisOSS\) At-Rest Encryption](#).
- ElastiCache (RedisOSS) (Modo de cluster ativado) versão 3.2.4

Se você precisar da funcionalidade do Redis OSS 2.8.x mais os seguintes recursos, escolha Redis OSS 3.2.4 (modo em 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 (RedisOSS) (modo sem cluster) 2.8.x e 3.2.4 (aprimorado)

Se o seguinte se aplicar a você, escolha Redis OSS 2.8.x ou Redis OSS 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 autoprotetidos, bem como para caches sem servidor.
- Você precisa oferecer suporte para vários bancos de dados.

## Resumo da comparação de Memcached, Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou OSS Redis (modo de cluster ativado)

	Memcached	Valkey ou Redis OSS (modo de cluster desativado)	Valkey ou Redis OSS (modo de cluster ativado)
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			
Alimentado RAMP	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 de cluster desativado)	Valkey ou Redis OSS (modo de cluster ativado)
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 classific ados	Não	Sim	Sim
Backup e restauração	Somente para Memcached sem servidor, não para clusters Memcached autoprojetados	Sim	Sim
Indexação geoespaci al	Não	4.0.10 e posterior	Sim

#### Observações:

string, objetos (como bancos de dados)

\* strings, conjuntos, conjuntos classificados, listas, hashes, bitmaps, hiperloglog

Memcached

Valkey ou Redis OSS  
(modo de cluster  
desativado)Valkey ou Redis OSS  
(modo de cluster  
ativado)

strings, conjuntos, conjuntos classificados, listas, hashes, bitmaps, hyperloglog, índices geoespaciais

+ 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 Valkey ou Redis OSS

Ao usar a migração on-line, você pode migrar seus dados do Valkey ou Redis de código aberto auto-hospedado na Amazon para a AmazonOSS. EC2 ElastiCache

### Note

A migração on-line não é compatível com caches ou clusters ElastiCache sem servidor executados no tipo de nó r6gd.

## Visão geral

Para migrar seus dados do Valkey ou do Redis de código aberto OSS em execução na Amazon para a EC2 Amazon, é ElastiCache necessária uma implantação da Amazon existente ou recém-criada. ElastiCache 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 a migração de dados do Valkey ou Redis de código aberto auto-hospedado OSS na EC2 Amazon ElastiCache para, e não para mover dados entre clusters. 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 a `StartMigration` API operação ou o AWS CLI comando. Ao migrar clusters desativados OSS no modo de cluster Valkey ou Redis, o processo de migração torna o nó primário do OSS cluster Valkey ou Redis uma réplica do ElastiCache Valkey ou do Redis primário de origem. OSS Ao migrar clusters habilitados para o OSS modo de cluster Valkey ou Redis, o processo de migração torna o nó primário de cada ElastiCache fragmento uma réplica do fragmento correspondente do cluster de origem que possui os mesmos slots.

Depois que as alterações do lado do cliente estiverem prontas, chame a `CompleteMigration` API operação. Essa API operação promove sua ElastiCache implantação em sua implantação primária do Valkey ou Redis OSS com nós primários e de réplica (conforme aplicável). Agora você pode redirecionar seu aplicativo cliente para começar a gravar ElastiCache dados no. Durante a migração, você pode verificar o status da replicação executando o INFO comando [valkey-cli](#) nos nós do Valkey e nos nós primários. ElastiCache

## Etapas da migração

Os seguintes tópicos descrevem o processo de migração dos seus dados:

- [Preparando sua origem e destino para a migração](#)
- [Testar a migração de dados](#)
- [Início da migração](#)
- [Verificação do andamento do processo de migração de dados](#)
- [Conclusão da migração de dados](#)

## Preparando sua origem e destino para a migração

Com essas etapas, você pode se preparar para migrar seus dados de uma fonte autônoma do Valkey ou do Redis EC2 para ElastiCache, ou de um cluster do Redis para um OSS cluster do Valkey. ElastiCache

Você deve garantir que todos os quatro pré-requisitos mencionados a seguir sejam atendidos antes de iniciar a migração do ElastiCache console ou. API AWS CLI

Para preparar seus Valkey ou Redis OSS Nodes de origem e destino para migração

1. Identifique a ElastiCache implantação de destino e certifique-se de que você possa migrar dados para ela.

Uma ElastiCache implantação existente ou recém-criada deve atender aos seguintes requisitos de migração:

- Ele está usando a versão 5.0.6 ou superior OSS do motor Valkey ou Redis.
  - Ele não tem criptografia em trânsito ativada.
  - Ele tem o multi-AZ habilitado.
  - Ele tem memória suficiente disponível para ajustar os dados do seu cluster Valkey ou RedisOSS. Para configurar as definições de memória reservada corretas, consulte [Gerenciando a memória reservada para Valkey e Redis OSS](#).
  - Para o modo de cluster desativado, você pode migrar diretamente das OSS versões 2.8.21 do Valkey ou Redis para o Valkey ou Redis da OSS versão 5.0.6 em diante se estiver usando o ou do Valkey ou Redis das versões 5.0.6 em diante usando o console ou. CLI OSS CLI Para o modo de cluster ativado, você pode migrar diretamente de qualquer versão Valkey ou Redis habilitada para o modo de cluster para a OSS versão 5.0.6 em diante do Redis, se estiver usando a OSS versão 5.0.6 ou posterior do Redis usando o console CLI ou. OSS CLI
  - 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 de seu Valkey ou Redis de código aberto OSS e a ElastiCache implantação sejam compatíveis.

No mínimo, todos os itens a seguir na ElastiCache implantação de destino devem ser compatíveis com sua OSS configuração Valkey ou Redis para replicação:

- Seu cluster não deveria ter sido AUTH ativado.
- A configuração `protected-mode` deve ser definida como. `no`
- Se você tiver uma `bind` configuração em sua OSS configuração do Valkey ou do Redis, ela deverá ser atualizada para permitir solicitações de nós. ElastiCache

- O número de bancos de dados lógicos deve ser o mesmo no ElastiCache nó e no cluster Valkey ou RedisOSS. Esse valor é definido usando databases a configuração do Valkey ou do RedisOSS.
  - OSSOs comandos Valkey ou Redis que realizam modificações de dados não devem ser renomeados para permitir que a replicação dos dados seja bem-sucedida. Por exemplo `sync,,,`, e `psync info config command cluster`
  - Para replicar os dados do seu OSS cluster Valkey ou Redis ElastiCache, verifique se há memória suficiente CPU para lidar com essa carga adicional. Essa carga vem do RDB arquivo criado pelo seu OSS cluster Valkey ou Redis e transferido pela rede para ElastiCache o nó.
  - Todas as OSS instâncias Valkey ou Redis no cluster de origem devem estar em execução na mesma porta.
3. Certifique-se de que suas instâncias possam se conectar ElastiCache fazendo o seguinte:
- Garanta que cada endereço IP da instância seja privado.
  - Atribua ou crie a ElastiCache implantação na mesma nuvem privada virtual (VPC) do Valkey ou do Redis OSS na sua instância (recomendado).
  - Se VPCs forem diferentes, configure o VPC peering para permitir o acesso entre os nós. Para obter mais informações sobre VPC emparelhamento, consulte [Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC](#).
  - O grupo de segurança vinculado às suas OSS instâncias Valkey ou Redis deve permitir o tráfego de entrada dos nós. ElastiCache
4. Certifique-se de que seu aplicativo possa direcionar o tráfego para ElastiCache os nós após a conclusão da migração dos dados. Para obter mais informações, consulte [Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC](#).

## Testar a migração de dados

Depois que todos os pré-requisitos forem concluídos, você poderá validar a configuração da migração usando o AWS Management Console, ou. ElastiCache API AWS CLI O exemplo a seguir mostra o uso doCLI.

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 usando CLI o.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache validará a configuração da migração sem nenhuma migração real de dados.

## Início da migração

Depois que todos os pré-requisitos forem concluídos, você poderá começar a migração de dados usando o AWS Management Console, ElastiCache API ou 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 o uso do CLI.

### Note

Recomendamos usar para `TestMigration` API 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 ElastiCache replicação de destino
- `--customer-node-endpoint-list`— Uma lista de endpoints com um DNS ou dois endereços IP e a porta em que seu OSS cluster Valkey ou Redis de origem 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 em cadeia, o endpoint pode apontar para uma réplica em vez do nó primário em seu cluster Valkey ou Redis. OSS

Veja a seguir um exemplo usando CLI o.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```



Conforme você executa esse comando, o nó ElastiCache primário (em cada fragmento) se configura para se tornar uma réplica da sua OSS instância do Valkey ou do Redis (no fragmento correspondente que possui os mesmos slots no redis habilitado para cluster). O status do ElastiCache cluster muda para migração e os dados começam a migrar da sua OSS instância Valkey ou Redis para o nó primário. ElastiCache Dependendo do tamanho dos dados e da carga em sua OSS instância Valkey ou Redis, a migração pode demorar um pouco para ser concluída. Você pode verificar o progresso da migração executando o INFO comando [valkey-cli na sua instância Valkey](#) e no nó primário. ElastiCache

Após a replicação bem-sucedida, todas as gravações em suas OSS instâncias Valkey ou Redis se propagam para o cluster. ElastiCache Você pode usar ElastiCache nós para leituras. No entanto, você não pode gravar no ElastiCache cluster. Se um nó ElastiCache primário tiver outros nós de réplica conectados a ele, esses nós de réplica continuarão a se replicar a partir do ElastiCache nó primário. Dessa forma, todos os dados do seu OSS cluster Valkey ou Redis são replicados em todos os nós do cluster. ElastiCache

Se um nó ElastiCache primário não puder se tornar uma réplica da sua OSS instância do Valkey ou do Redis, ele tentará novamente várias vezes antes de finalmente voltar a ser primário. O status do ElastiCache cluster então muda para disponível e um evento do grupo de replicação sobre a falha em iniciar a migração é enviado. 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 [Preparando sua origem e destino para a migração](#) foram seguidas.
- Certifique-se de que a configuração de roteamento da sua VPC e das sub-redes permita o tráfego entre os ElastiCache nós e suas instâncias Valkey ou Redis. OSS
- Certifique-se de que o grupo de segurança anexado às suas OSS instâncias Valkey ou Redis permita tráfego vinculado à entrada dos ElastiCache nós.
- Verifique os OSS registros do Valkey ou do Redis de suas instâncias para obter mais informações sobre falhas específicas da 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 Valkey ou o OSS `master_link_status` Redis estão no INFO comando up no (s) nó (s) ElastiCache primário (s). Você também pode encontrar essas informações no ElastiCache console. Selecione o cluster e, em CloudWatch métricas, observe Primary Link Health Status. Depois que o valor atinge 1, os dados estão sincronizados.
- Você pode verificar se a ElastiCache réplica tem um estado on-line executando o INFO comando em suas instâncias do Valkey ou do RedisOSS. Isso também fornece informações sobre o atraso da replicação.
- Verifique o baixo buffer de saída do cliente usando o [CLIENTLIST](#) comando em suas instâncias Valkey ou RedisOSS.

Depois que a migração de dados for concluída, os dados estarão sincronizados com qualquer nova gravação que chegue ao (s) nó (s) primário (s) do seu cluster Valkey ou RedisOSS.

## Conclusão da migração de dados

Quando você estiver pronto para migrar para o ElastiCache cluster, use o `complete-migration` CLI comando 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
```

Conforme você executa esse comando, o nó ElastiCache primário (em cada fragmento) interrompe a replicação da sua OSS instância do Valkey ou do Redis e a promove para primária. Normalmente, essa promoção é concluída em alguns minutos. Para confirmar a promoção para primário, verifique o evento `Complete Migration successful for test-cluster`. Nesse ponto, você pode direcionar seu aplicativo para ElastiCache gravações e leituras. ElastiCache o status do cluster deve mudar de migrante para disponível.

Se a promoção para primário falhar, o nó ElastiCache primário continuará a se replicar a partir da sua instância Valkey ou RedisOSS. O ElastiCache cluster continua em status de migração e uma mensagem de evento do grupo de replicação sobre a falha é enviada. 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. Nesse caso, certifique-se de que o ElastiCache primário possa ser replicado da sua OSS instância Valkey ou Redis e que ambos estejam sincronizados. 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 ElastiCache nós estiver sendo substituído. 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

Você pode usar o AWS Management Console para migrar seus dados do cluster para o cluster Valkey ou RedisOSS.

Como executar a migração de dados online usando o console

1. Faça login no console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Crie um novo cluster Valkey ou Redis ou escolha um OSS cluster existente. Verifique se o cluster atende aos seguintes requisitos:
  - A versão do seu motor deve ser Valkey 7.2 e superior, ou Redis OSS 5.0.6 ou superior.
  - Seu cluster não deveria ter sido AUTH ativado.
  - A configuração `protected-mode` deve ser definida como `no`.
  - Se você tiver uma `bind` configuração em sua OSS configuração do Valkey ou do Redis, ela deverá ser atualizada para permitir solicitações de nós. ElastiCache
  - O número de bancos de dados deve ser o mesmo entre o ElastiCache nó e seu cluster Valkey ou RedisOSS. Esse valor é definido usando `databases` a configuração do mecanismo.
  - OSSOs comandos Valkey ou Redis que realizam modificações de dados não devem ser renomeados para permitir que a replicação dos dados seja bem-sucedida.
  - Para replicar os dados do seu OSS cluster Valkey ou Redis ElastiCache, verifique se há memória suficiente CPU para lidar com essa carga adicional. Essa carga vem do RDB arquivo criado pelo seu OSS cluster Valkey ou Redis e transferido pela rede para ElastiCache o nó.
  - 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 em que seu cluster Valkey ou Redis está disponível OSS.

 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.
```

## Escolhendo regiões e zonas de disponibilidade para ElastiCache

Você pode fornecer escalabilidade e confiabilidade adicionais aos seus ElastiCache clusters designando regiões e zonas de disponibilidade usando o endpoint correspondente.

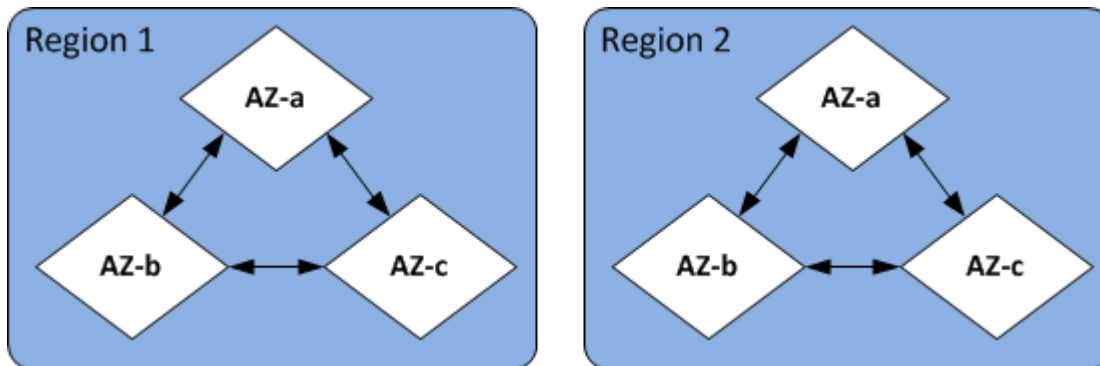
AWS Os recursos de computação em nuvem estão alojados em instalações de data center altamente disponíveis. Para fornecer escalabilidade e confiabilidade adicionais, estas instalações do datacenter estão localizadas em diferentes locais físicos. Esses locais são categorizados por regiões e zonas de disponibilidade.

AWS As regiões são grandes e amplamente dispersas em localizações geográficas separadas. As zonas de disponibilidade são locais distintos dentro de uma AWS região que são projetados para serem isolados de falhas em outras zonas de disponibilidade. Eles fornecem conectividade de rede barata e de baixa latência para outras zonas de disponibilidade na mesma AWS região.

**⚠ Important**

Cada região é totalmente independente. Qualquer ElastiCache atividade que você iniciar (por exemplo, criar clusters) é executada somente na sua 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

### Tópicos

- [Considerações sobre a zona de disponibilidade com o Memcached](#)
- [Localização dos seus nós](#)
- [Regiões e endpoints com suporte](#)
- [Usando zonas locais com ElastiCache](#)
- [Usando Outposts com ElastiCache](#)

## Considerações sobre a zona 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

ElastiCache o armazenamento em cache sem servidor cria um cache altamente disponível que abrange várias zonas de disponibilidade. Você pode especificar sub-redes de diferentes zonas de disponibilidade e da VPC mesma forma que cria seu cluster sem servidor ou ElastiCache escolherá sub-redes automaticamente a partir do padrão. VPC

### Projetando seu próprio ElastiCache cluster (Memcached)

Um cluster do Memcached pode ter até 300 nós. Ao criar ou adicionar nós ao seu cluster Memcached, você pode especificar uma única zona de disponibilidade para todos os seus nós, permitir ElastiCache a escolha de uma única zona de disponibilidade para todos os seus nós, especificar as zonas de disponibilidade para cada nó ou permitir ElastiCache a escolha de 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 de zona de disponibilidade tenha seus nós distribuídos em várias zonas de disponibilidade, 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 atualmente ativos 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 ElastiCache determinar 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`) para o número de nós de cache ativos que você deseja após a aplicação dessa modificação.
  - Defina `CacheNodeIdsToRemove` (CLI:- `-nodes-to-remove`) como uma lista dos nós de cache que você deseja remover do cluster.

O número de nós de cache IDs listados deve ser igual ao número de nós atualmente ativos menos o valor em `NumCacheNodes`.

- (Opcional) Defina `ApplyImmediately` (CLI:- `-apply-immediately`) como verdadeiro.

Se você não definir `ApplyImmediately` (CLI:- `-apply-immediately`) como `true`, as exclusões dos nós ocorrerão na próxima janela de manutenção.

## Localização dos seus nós

A Amazon ElastiCache oferece suporte à localização de todos os nós de um cluster em uma ou várias zonas de disponibilidade (AZs). Além disso, se você optar por localizar seus nós em vários AZs (recomendado), ElastiCache permite que você escolha a AZ para cada nó ou permita ElastiCache escolhê-los 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 cause a falha de todo o sistema. Os testes demonstraram que não há diferença significativa de latência entre localizar todos os nós em uma AZ ou distribuí-los por vários. AZs

Você pode especificar uma AZ para cada nó ao criar um cluster ou adicionar nós ao modificar um cluster existente. Para obter mais informações, consulte as informações a seguir.

- [Criação de um cluster para Memcached](#)
- [Criação de um cluster para Valkey ou Redis OSS](#)
- [Modificando um cluster ElastiCache](#)
- [Adicionando nós a um ElastiCache cluster](#)

## Regiões e endpoints com suporte

A Amazon ElastiCache está disponível em várias AWS regiões. Isso significa que você pode lançar ElastiCache clusters em locais que atendam às suas necessidades. Por exemplo, você pode lançar na AWS região mais próxima de seus clientes ou em uma AWS região específica 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). ElastiCache Os caches sem servidor replicam automaticamente os dados em várias zonas de disponibilidade (exceto us-west-1 quando os dados são replicados em duas zonas de disponibilidade) para obter alta disponibilidade. Ao projetar seu próprio ElastiCache cluster, você pode optar por iniciar seus nós de forma diferente AZs para obter tolerância a falhas. Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Escolhendo regiões e zonas de disponibilidade para ElastiCache](#) no início deste tópico.



## Regiões onde ElastiCache é suportado

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 (N. 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
Asia Pacific (Sydney) Region 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 (UAE) 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 da China (Pequim) cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS	
Região da 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 como usar o AWS GovCloud (EUA) com ElastiCache, consulte [Serviços na região 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 AWS região, consulte [Tipos de nós compatíveis por região da AWS](#).

Para ver uma tabela de AWS produtos e serviços por região, consulte [Produtos e serviços por região](#).

## Usando zonas locais com ElastiCache

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 uma Zona Local criando uma nova sub-rede e atribuindo-a à Zona Local. Quando você cria uma sub-rede

em uma zona local, ela VPC é estendida para essa zona local. A sub-rede na zona local opera da mesma forma que as outras sub-redes na sua VPC.

Ao usar as Zonas Locais, você pode colocar recursos como um ElastiCache cluster em vários locais próximos aos seus usuários.

Ao criar um ElastiCache cluster, você pode escolher uma sub-rede em uma zona local. As zonas locais têm suas próprias conexões com a Internet e suporte no AWS 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 AWS região seguido por um identificador que indica a localização, 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 às ElastiCache 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 M5: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

Tipos de nó R5: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

Tipos de nó T3: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

## Ativação de uma zona local

1. Ative a zona local no EC2 console da Amazon.

Para obter mais informações, consulte [Enabling Local Zones](#) no Amazon EC2 User Guide.

2. Crie uma sub-rede na zona local.

Para obter mais informações, consulte [Criação de uma sub-rede VPC no](#) Guia do VPC usuário da Amazon.

### 3. Crie um grupo de ElastiCache sub-redes na zona local.

Ao criar um grupo de ElastiCache sub-redes, escolha o grupo de zonas de disponibilidade para a zona local.

Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

### 4. Crie um cluster ElastiCache (Memcached) que use a ElastiCache sub-rede na zona local.

Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).

### 5. Crie um cluster ElastiCache (RedisOSS) que use a ElastiCache sub-rede na zona local. Para obter mais informações, consulte um dos tópicos a seguir:

- [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)
- [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

## 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 registros do motor e os registros lentos não podem ser ativados.
- ElastiCache on Outposts não suporta CoIP.

- 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 AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha caches Valkey, caches Redis ou caches OSS Memcached.
3. Se você escolheu caches Valkey, selecione Criar cache Valkey. Se você escolheu OSScaches Redis, selecione Criar cache 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 Desativado. Em seguida, crie um nome e uma descrição opcional para o cache.
5. Para localização, escolha No local.
6. Na seção Local, você verá o campo Outpost ID. 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 a VPCID. Deixe o resto como padrão e selecione Avançar.

### Configurar opções on-premises

Você pode selecionar um Outpost disponível para adicionar seu cluster de cache ou, se não houver Outposts disponíveis, criar um novo usando as seguintes etapas:

Em On-Premises options (Opções on-premises):

1. Nas configurações do Valkey, configurações do Redis ou OSS 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 RedisOSS, aceite a porta padrão 6379. Para Memcached, aceite a porta padrão 11211. Se você 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 Valkey ou RedisOSS, Porting Assistant for .NET for Outposts oferece suporte apenas às 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 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.
  - VPCID: O VPC ID deve corresponder ao Posto AvançadoVPC. Se você selecionar uma VPC que não tenha sub-rede IDs nos Outposts, 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 (Nome de recurso da Amazon) e o número do AWS recurso.
- Nome do Posto Avançado: O nome do Posto AWS Avançado.

## Usando Outposts com o AWS CLI

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 o AWS CLI para operações ad hoc (únicas).

### Baixando e configurando o AWS CLI

AWS CLI É executado em Windows, macOS ou Linux. Use o procedimento a seguir para fazer download e configurá-la.

Para baixar, instalar e configurar o CLI

1. Faça o download AWS CLI na página da Web da [interface de linha de AWS comando](#).
2. Siga as instruções para [instalar AWS CLI](#) e [configurar o AWS CLI](#) no Guia do AWS Command Line Interface usuário.

### Usando o AWS CLI com Outposts

Use a CLI operação a seguir para criar um cluster de cache que usa Outposts:

- [create-cache-cluster](#)— Usando essa operação, o `outpost-mode` parâmetro aceita um valor que especifica se os nós no cluster de cache 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 cache 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](#)
- [ElastiCache melhores práticas e estratégias de armazenamento em cache](#)
- [Gerenciando seu cluster autoprojeto em ElastiCache](#)
- [Dimensionamento ElastiCache](#)
- [Introdução ao JSON Valkey e ao Redis OSS](#)
- [Marcando seus recursos ElastiCache](#)
- [Usando o Amazon ElastiCache Well-Architected Lens](#)
- [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 seus dados criando um novo cache Valkey, Redis OSS ou Serverless Memcached e preenchendo-o com dados de um backup. Com ElastiCache, você pode gerenciar backups usando o AWS Management Console, the AWS Command Line Interface (AWS CLI) ElastiCache API e.

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 autoprotetidos](#)
- [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: semeando um novo cluster autoprotetido com um backup criado externamente](#)

## Restrições do backup

Considere as seguintes restrições ao planejar ou fazer backups:

- O backup e a restauração são suportados somente para caches executados em Valkey, Redis OSS ou Serverless Memcached.
- Para clusters Valkey ou Redis OSS (modo de cluster desativado), o backup e a restauração não são suportados nos `cache.t1.micro` nós. Todos os outros tipos de nó de cache têm suporte.
- Para clusters Valkey ou Redis OSS (modo de cluster ativado), o backup e a restauração são compatíveis com todos os tipos de nós.
- Durante qualquer período contíguo de 24 horas, você pode criar no máximo 24 backups manuais por cache sem servidor. Para clusters OSS autoprotetidos pelo Valkey e Redis, você pode criar no máximo 20 backups manuais por nó no cluster.
- O Valkey ou o Redis OSS (modo de cluster ativado) só suportam a realização de backups no nível do cluster (para o API ou CLI, no nível do grupo de replicação). O Valkey ou o Redis OSS (modo de cluster ativado) não suportam a realização de backups no nível do fragmento (para o API ou CLI, no nível do grupo de nós).
- Durante o processo de backup, você não pode executar nenhuma outra CLI operação API ou operação no cache sem servidor. Você pode executar API e suas CLI operações em um cluster projetado por você mesmo durante o backup.

- Se você estiver usando OSS caches Valkey ou Redis com hierarquização de dados, não poderá 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 autoprotetidos

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 autoprotetidos, é possível que haja algum impacto sobre o desempenho, dependendo da memória reservada disponível. Os backups para clusters autoprotetidos não estão disponíveis com ElastiCache (Memcached), mas estão disponíveis com ElastiCache (Redis). OSS

Estas são as diretrizes a fim de melhorar o desempenho do backup para clusters autoprotetidos.

- 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 o 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 [Gerenciando a memória reservada para Valkey e Redis OSS](#)
- Crie backups a partir de uma réplica de leitura — Se você estiver executando o Valkey ou o Redis OSS em um grupo de nós com mais de um nó, poderá fazer um backup do nó primário ou de uma das réplicas de leitura. Devido aos recursos do sistema necessários durante esse período BGSAVE, recomendamos que você crie backups a partir de uma das réplicas de leitura. Enquanto o backup está sendo criado a partir da réplica, o nó primário permanece inalterado pelos requisitos de BGSAVE recursos. 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 OSS dados mais recentes do Valkey ou do Redis, antes que o grupo de replicação seja excluído.

## Programação de backups automáticos

Você pode habilitar backups automáticos para qualquer cache OSS sem servidor Valkey ou Redis ou cluster autoprojeto. 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](#).

Você pode ativar backups automáticos para qualquer cache Memcached Serverless. 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 ElastiCache API o. Para Valkey e RedisOSS, isso é feito marcando a caixa Ativar backups automáticos na seção Configurações avançadas do Valkey ou Configurações avançadas do Redis OSS. Para o 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 o AWS Management Console AWS CLI, o ou ElastiCache API o.

### Criação de um backup manual (console)

Para criar um backup de um cache (console)

1. Faça login no AWS Management Console e abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação, escolha caches Valkey, caches Redis ou OSS caches Memcached, dependendo da sua preferência.
3. Escolha 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 hifens 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 projetado por você mesmo com o AWS CLI

Para criar um backup manual de um cluster autoprotetado 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 esse parâmetro somente ao fazer backup de um cluster Valkey ou Redis OSS (modo de cluster desativado).

- `--replication-group-id`— Nome do cluster Valkey ou Redis OSS (modo de cluster ativado) (CLI/API: um grupo de replicação) a ser usado como fonte para o backup. Use esse parâmetro ao fazer backup de um cluster Valkey ou Redis OSS (modo de cluster ativado).
- `--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: Fazendo backup de um cluster Valkey ou Redis OSS (modo de cluster desativado) que não tem nós de réplica

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: Fazendo backup de um cluster Valkey ou Redis OSS (modo de cluster desativado) com nós de réplica

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
```

## Exemplo de saída: fazendo backup de um cluster Valkey ou Redis OSS (modo de cluster desativado) com nós de réplica

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: Fazendo backup de um cluster para Valkey ou Redis OSS (modo de cluster ativado)

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.

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

```

### Exemplo de saída: Fazendo backup de um cluster Valkey ou Redis OSS (modo de cluster ativado)

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 da

Para obter mais informações, consulte [create-snapshot](#) na Referência de comandos da AWS CLI .

## Criação de um backup final

Você pode criar um backup final usando o ElastiCache console AWS CLI, o ou ElastiCache API o.

### Criação de um backup final (console)

Você pode criar um backup final ao excluir um cache OSS sem servidor Valkey ou Redis, um cluster OSS autoprojeto pelo Valkey ou Redis ou um cache sem servidor Memcached usando o console. ElastiCache

Para criar um backup final ao excluir um cache, na caixa de diálogo de exclusão, escolha Sim em Criar backup e dê um nome ao backup.

Tópicos relacionados da

- [Usando o AWS Management Console](#)
- [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 o. AWS CLI

Tópicos

- [Ao excluir um cache Valkey, cache Redis ou OSS cache sem servidor Memcached](#)
- [Ao excluir um OSS cluster Valkey ou Redis sem réplicas de leitura](#)
- [Ao excluir um OSS cluster Valkey ou Redis com réplicas de leitura](#)

Ao excluir um cache Valkey, cache Redis ou OSS cache sem servidor Memcached

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 AWS CLI Comandos.

Ao excluir um OSS cluster Valkey ou Redis sem réplicas de leitura

Para criar um backup final para um cluster autoprotetido 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 AWS CLI Comandos.

## Ao excluir um OSS cluster Valkey ou Redis 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 AWS CLI Comandos.

## 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 AWS Management Console

1. Faça login no AWS Management Console 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, opcionalmente, detalhes sobre um backup específico, use a operação. `describe-serverless-cache-snapshots` 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.

## Descrição dos backups de cluster autoprotetidos (AWS CLI)

Para exibir uma lista de backups de cluster autoprotetidos e, opcionalmente, detalhes sobre um backup específico, use a `describe-snapshots` CLI operação.

### 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 AWS Management Console 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.

Cópia de um backup de cluster autoprotetido (AWS CLI)

Para copiar um backup de um cluster autoprotetido, 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 my-backup-copy
```

Para Windows:

```
aws elasticache copy-snapshot ^
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^
  --target-snapshot-name my-backup-copy
```

Para ter mais informações, consulte [copy-snapshot](#) no AWS CLI.

## Exportação de um backup

A Amazon ElastiCache suporta a exportação do seu backup ElastiCache (RedisOSS) 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 ElastiCache API o.

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: semeando um novo cluster autoprojeto com um backup criado externamente](#). Outro motivo pelo qual você pode querer exportar os dados do seu cache é 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 [Hierarquização de dados em ElastiCache](#).
- A exportação de um backup está disponível para clusters OSS autoprojeto do Valkey e do Redis, Serverless Valkey and Redis e Serverless Memcached. OSS A exportação de um backup não está disponível para clusters Memcached autoprojeto.

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.

### Criar um bucket do 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 AWS Management Console 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 Nome do bucket, digite um nome para o bucket do Amazon S3.

O nome do seu bucket do Amazon S3 deve ser DNS compatível. Caso contrário, não ElastiCache conseguirá acessar seu arquivo de backup. As regras de DNS conformidade 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 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, é altamente recomendável que você configure IAM políticas 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 AWS Management Console 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 [Criar um bucket do Amazon S3](#).
3. Escolha a guia Permissões e, em Permissões, escolha Lista de controle de acesso (ACL) e escolha Editar.
4. Adicionar ID canônico do beneficiário  
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353 com as seguintes opções:
  - Objetos: listar, gravar
  - BucketACL: leitura, gravação

 Note

- Para a PDT GovCloud região, o ID canônico é.  
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Para a OSU GovCloud região, 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 AWS CLI, o ou o ElastiCache API para exportar seu snapshot para ele. Os exemplos a seguir pressupõem que a IAM identidade do chamador tenha as seguintes permissões adicionais específicas IAM do S3.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets",
      "s3:PutObject",
      "s3:GetObject",
      "s3:DeleteObject",
      "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::*"
  }]
}
```

Para escolher regiões, veja a seguir um exemplo da aparência da política atualizada para o bucket do S3. O exemplo usa a região Ásia-Pacífico (Hong Kong).

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticache.amazonaws.com"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    },
    {
      "Sid": "Stmt15399484",
      "Effect": "Allow",
      "Principal": {
        "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
      },
      "Action": "s3:*",
```

```
    "Resource": [
      "arn:aws:s3:::hkg-elasticache-backup",
      "arn:aws:s3:::hkg-elasticache-backup/*"
    ]
  }
]
```

## 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 AWS Management Console 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 [Criar um bucket do 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 READ permissões %s no S3 Bucket.

Solução: adicione permissões de Leitura no bucket.

- ElastiCache não recebeu WRITE permissões %s no S3 Bucket.

Solução: adicione permissões de Gravação no bucket.

- ElastiCache não recebeu READ \_ ACP permissões %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 `export-serverless-cache-snapshot` CLI operação 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
```

Exportação de um backup de ElastiCache cluster projetado por você mesmo ( )AWS CLI

Exportação de um backup de um cluster autoprojeto

Exporte o backup para um bucket do Amazon S3 usando a `copy-snapshot` CLI operação 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 autoprotetado e restaurar um OSS backup existente do Redis em um novo cache do Redis ou cluster autoprotetado. Você também pode restaurar um backup existente do cache sem servidor do Memcached em um novo cache sem servidor do Memcached.

### Restauração de um backup para um cache sem servidor (console)

#### Note

ElastiCache O Serverless suporta RDB arquivos compatíveis com Valkey 7.2 e superior e OSS versões do Redis entre 5.0 e a versão mais recente disponível.

### Para restaurar um backup para um cache sem servidor (console)

1. Faça login no AWS Management Console 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 autoprotetado (console)

#### Para restaurar um backup para um cluster autoprotetado (console)

1. Faça login no AWS Management Console 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 Crie seu próprio cache 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 cache autoprojeto e importe dados do backup.

### Restauração de um backup para um cache sem servidor (AWS CLI)

#### Note

ElastiCache O Serverless suporta RDB arquivos compatíveis com Valkey 7.2 e superior e OSS versões do Redis 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
```

### Restauração de um backup para um cluster autoprojeto (AWS CLI)

Para restaurar um backup para um cluster autoprojeto (AWS CLI)

Você pode restaurar um backup Valkey ou Redis OSS (modo de cluster desativado) de duas maneiras.

- ```
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
```

## Restauração de um backup para um cluster autoprotetado (AWS CLI)

### Para restaurar um backup para um cluster autoprotetado (AWS CLI)

Você pode restaurar um backup de cache OSS sem servidor Valkey ou Redis e também pode restaurar um cluster autoprotetado pelo Valkey ou Redis. OSS

Você pode restaurar um backup de cache OSS sem servidor Valkey ou Redis de duas maneiras.

- Você pode restaurar para um cluster Valkey ou Redis de nó único OSS (modo de cluster desativado) usando a operação. AWS CLI `create-cache-cluster`
- Você pode restaurar em um OSS cluster Valkey ou Redis com réplicas de leitura (um grupo de replicação). Para fazer isso, você pode usar Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado) com a operação. AWS CLI `create-replication-group` Nesse caso, você semeia a restauração com um arquivo.rdb do Valkey ou do RedisOSS. Para obter mais informações sobre como propagar um novo cluster autoprotetado, consulte [Tutorial: semeando um novo cluster autoprotetado com um backup criado externamente](#).

Você pode restaurar um backup Valkey ou Redis OSS (modo de cluster desativado) de duas maneiras.

- Você pode restaurar para um cluster Valkey ou Redis de nó único OSS (modo de cluster desativado) usando a operação. AWS CLI `create-cache-cluster`
- Você pode restaurar em um OSS cluster Valkey ou Redis com réplicas de leitura (um grupo de replicação). Para fazer isso, você pode usar Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado) com a operação. AWS CLI `create-replication-group` Nesse caso, você semeia a restauração com um arquivo.rdb do Valkey ou do RedisOSS. Para obter mais informações sobre como propagar um novo cluster autoprotetado, consulte [Tutorial: semeando um novo cluster autoprotetado com um backup criado externamente](#).

Ao usar a operação `create-cache-cluster` ou `create-replication-group`, não se esqueça de incluir o parâmetro `--snapshot-name` ou `--snapshot-arn` para propagar o novo cluster ou grupo de replicação com os dados do backup.

## 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 API operação 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 ElastiCache API o.

### Exclusão de um backup (console)

O procedimento a seguir exclui um backup usando o ElastiCache console.

Para excluir um backup

1. Faça login no AWS Management Console 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 AWS CLI Comandos.

Exclusão de um backup de cluster autoprotetido (AWS CLI)

Use a AWS CLI operação `delete-snapshot` com o parâmetro a seguir para excluir um backup de cluster projetado por você mesmo.

- `--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](#).

As etiquetas de alocação de custos são um meio de rastrear seus custos em vários AWS serviços, agrupando suas despesas em faturas por valores de etiquetas. Para saber mais sobre alocação de custos, consulte [Usar tags de alocação de custos](#).

Usando o ElastiCache console, o AWS CLI, ou ElastiCache API você pode adicionar, listar, modificar, remover ou copiar etiquetas de alocação de custos em seus backups. Para obter mais informações, consulte [Monitoramento de custos com tags de alocação de custos](#).

## Tutorial: semeando um novo cluster autoprojetoado com um backup criado externamente

Ao criar um novo cluster OSS autoprojetoado pelo Valkey ou Redis, você pode semeá-lo com dados de um arquivo de backup Valkey ou OSS Redis .rdb. A semeadura do cluster é útil se você atualmente gerencia uma OSS instância do Valkey ou do Redis fora dela ElastiCache e deseja preencher seu novo cluster autoprojetoado ElastiCache (RedisOSS) com seus dados existentes do Valkey ou do Redis. OSS

Para implantar um novo cluster OSS autoprojetoado do Valkey ou do Redis a partir de um backup do Valkey ou do Redis OSS criado na Amazon, consulte. ElastiCache [Restauração de um backup para um novo cache](#)

Ao usar um arquivo Valkey ou Redis OSS .rdb para semear um novo cluster autoprojetoado, você pode fazer o seguinte:

- Faça o upgrade de um cluster não particionado para um cluster autoprojetoado pelo Valkey ou Redis OSS (modo de cluster ativado) executando o Redis versão 3.2.4. OSS
- Especifique vários fragmentos (chamados de grupos de nós no API eCLI) no novo cluster autoprojetoado. Esse número pode ser diferente do número de fragmentos no cluster autoprojetoado que foi usado para criar o arquivo de backup.
- Especifique um tipo de nó diferente para o novo cluster autoprojetoado, 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 do RedisOSS. Para obter mais informações, consulte [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#).
- Distribua suas chaves nos slots do novo cluster Valkey ou Redis OSS (modo de cluster ativado) de forma diferente do que no cluster usado para criar o arquivo de backup.

### Note

Você não pode semear um cluster Valkey ou Redis OSS (modo de cluster desativado) a partir de um arquivo.rdb criado a partir de um cluster Valkey ou Redis OSS (modo de cluster ativado).

**⚠ Important**

- Você deve garantir que seus dados de OSS backup do Valkey ou do Redis não excedam os recursos do nó. Por exemplo, você não pode fazer upload de um arquivo.rdb com 5 GB de OSS dados do Valkey ou do Redis para 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 do tipo de OSS nó Redis e detalhes dos ElastiCache produtos da Amazon](#).

- Você pode criptografar um arquivo Valkey ou Redis OSS .rdb somente com a criptografia do lado do servidor (-S3) do Amazon S3. SSE Para obter mais informações, consulte [Proteger dados usando a criptografia no lado do servidor](#).

A seguir, você encontrará tópicos que orientam você na migração de seu cluster de fora ElastiCache para Valkey ou Redis OSS para ElastiCache (Redis). OSS

#### Migrando para ElastiCache (RedisOSS)

- [Etapa 1: criar um backup 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 para ElastiCache (RedisOSS).

- [Etapa 1: criar um backup 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 Valkey ou Redis OSS

Para criar o OSS backup do Valkey ou do Redis para alimentar sua instância ElastiCache (RedisOSS)

1. Conecte-se à sua instância existente do Valkey ou do RedisOSS.
2. Execute uma BGSAVE ou outra SAVE operação 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](#)o site da Valkey.

SAVE é síncrono e bloqueia outros processos até terminar. Para obter mais informações, consulte [SAVE](#)o site da Valkey.

Para obter informações adicionais sobre a criação de um backup, consulte [Persistência](#) no site da 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 AWS Management Console 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 seu bucket do Amazon S3 deve ser DNS compatível. Caso contrário, não ElastiCache conseguirá acessar seu arquivo de backup. As regras de DNS conformidade 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 ElastiCache (RedisOSS). 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 sejam usados para semear seu novo cluster Valkey ou RedisOSS.

Para adicionar uma pasta a um bucket do Amazon S3

1. Faça login no AWS Management Console 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 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 AWS Management Console 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 AWS Management Console 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 do 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 de leitura/gravação de objetos ACL
  - 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 AWS Management Console 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.

```
{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Principal": {
        "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::example-bucket",
        "arn:aws:s3:::example-bucket/backup1.rdb",
        "arn:aws:s3:::example-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 para Valkey ou Redis OSS](#) ou [Criando um grupo de OSS replicação Valkey ou Redis do zero](#). Certifique-se de escolher Valkey ou Redis OSS como seu mecanismo de cluster.

O método que você usa para saber ElastiCache onde encontrar o backup que você enviou para o Amazon S3 depende do método usado para criar o cluster:

Semeie o cluster ElastiCache (RedisOSS) ou grupo de replicação 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 Localização RDB do arquivo semente S3, digite o caminho do Amazon S3 para os arquivos. 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 create-replication-group operação create-cache-cluster ou, use o parâmetro --snapshot-arns para especificar um totalmente qualificado ARN para cada arquivo.rdb. Por exemplo, arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb. Eles ARN devem ser resolvidos nos arquivos de backup que você armazenou no Amazon S3.

- Usando o ElastiCache API

Se você usar a CreateReplicationGroup ElastiCache API operação CreateCacheCluster ou, use o parâmetro SnapshotArns para especificar um totalmente qualificado ARN para cada arquivo.rdb. Por exemplo, arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb. Eles ARN devem ser resolvidos nos arquivos de backup que você armazenou no Amazon S3.

### Important

Ao semear um cluster Valkey ou Redis OSS (modo de cluster ativado), você deve 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 obter mais informações, consulte as informações a seguir.

- CLI: [create-replication-group](#)na AWS CLI referência
- API: [CreateReplicationGroup](#)na ElastiCache API referência

Durante o processo de criação do cluster, os dados no OSS backup do Valkey ou do Redis 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 da linha de AWS ElastiCache comando ou ElastiCache API 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, Redis OSS e Memcached compatíveis e como fazer o upgrade. Observe que todos os recursos disponíveis com o Redis OSS 7.2 estão disponíveis no Valkey 7.2 e superior por padrão. Você também pode fazer o upgrade de alguns OSS mecanismos existentes ElastiCache com Redis para um motor Valkey.

## Tópicos

- [Gerenciamento de versões para ElastiCache](#)
- [Como atualizar as versões dos mecanismos](#)
- [Como acionar a atualização entre mecanismos do Redis OSS para o Valkey](#)
- [Mecanismos e versões compatíveis](#)
- [Principais diferenças de comportamento e compatibilidade da versão com o Valkey](#)
- [Principais diferenças de comportamento e compatibilidade da versão com o Redis OSS](#)
- [Resolvendo atualizações bloqueadas do motor Valkey ou Redis OSS](#)

## Gerenciamento de versões para ElastiCache

Gerencie como você gostaria de atualizar seus ElastiCache caches e clusters autoprotetidos atualizados para os mecanismos Valkey, Redis OSS e Memcached.

### 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 MINOR e a versão do PATCH software ao seu cache, sem nenhum impacto ou tempo de inatividade em seu aplicativo. Não é necessária nenhuma ação de sua parte.

Quando uma nova MAJOR versão estiver disponível, o ElastiCache Serverless enviará uma notificação no console e um evento no. EventBridge Você pode optar por atualizar seu cache para a versão principal mais recente modificando seu cache usando o console ou API selecionando a versão mais recente do mecanismo. CLI

## Gerenciamento de versões para clusters autoprotetidos ElastiCache

Ao trabalhar com ElastiCache clusters autoprotetidos, você pode controlar quando o software que alimenta seu cluster de cache é atualizado para novas versões suportadas pelo. ElastiCache Você pode controlar quando atualizar seu cache para as PATCH versões e versões mais recentes disponíveis MAJOR. MINOR 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 de cache 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 OSS cluster Valkey ou Redis for replicado em uma ou mais regiões, a versão do mecanismo será atualizada para regiões secundárias e depois para a região primária.
- ElastiCache As versões (RedisOSS) são identificadas com uma versão semântica que compreende um componente e. MAJOR MINOR Por exemplo, no Redis OSS 6.2, a versão principal é 6 e a versão secundária 2. Ao operar clusters autoprotetidos, ElastiCache (RedisOSS) também expõe o PATCH componente, por exemplo, Redis OSS 6.2.1, e a versão do patch é 1.

MAJOR as versões são para alterações API incompatíveis e as MINOR versões são para novas funcionalidades adicionadas de forma compatível com versões anteriores. PATCH as versões são para correções de bugs compatíveis com versões anteriores e alterações não funcionais.

Com o Valkey e o RedisOSS, você inicia os upgrades da versão do mecanismo em 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 fazer o upgrade para uma versão mais recente, você deve modificar seu cluster de cache 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 [Modificando um cluster 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](#).

## Considerações sobre atualização ao trabalhar com clusters autoprotetidos

### Note

As considerações a seguir só se aplicam ao atualizar clusters autoprotetidos. Eles não se aplicam ao ElastiCache Serverless.

## Considerações sobre Valkey e Redis OSS

Ao atualizar um OSS cluster Valkey ou Redis autoprotetido, 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.
- Começando com o Valkey 7.2 e o Redis OSS 6.0, ElastiCache oferecerá uma única versão para cada lançamento menor, em vez de oferecer várias versões de patch.



- A partir da versão 5.0.6 OSS do mecanismo Redis, você pode atualizar sua versão de 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 levar a um tempo maior de failover durante a DNS propagação (30s-1m).
- 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 (RedisOSS) foi projetado para fazer o melhor esforço para reter seus dados existentes e requer uma replicação bem-sucedida do RedisOSS.
- Ao atualizar o mecanismo, ElastiCache encerrará as conexões existentes do cliente. [Para minimizar o tempo de inatividade durante as atualizações do mecanismo, recomendamos que você implemente as melhores práticas para OSS clientes Redis com novas tentativas de erro e recuo exponencial, além das melhores práticas para minimizar o tempo de inatividade durante a manutenção.](#)
- Você não pode atualizar diretamente do Valkey ou do Redis OSS (modo de cluster desativado) para o Valkey ou Redis OSS (modo de cluster ativado) ao atualizar seu mecanismo. O procedimento a seguir mostra como fazer o upgrade do Valkey ou do Redis OSS (modo de cluster desativado) para o Valkey ou Redis OSS (modo de cluster ativado).

Para atualizar de uma versão do mecanismo Valkey ou Redis OSS (modo de cluster desativado) para Valkey ou Redis OSS (modo de cluster ativado)

1. Faça um backup do seu cluster ou grupo de replicação Valkey ou Redis OSS (modo de cluster desativado). Para obter mais informações, consulte [Realização de backups manuais](#).
2. Use o backup para criar e semear um cluster Valkey ou Redis OSS (modo de cluster ativado) 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: semeando um novo cluster autoprojeto com um backup criado externamente](#).
3. Exclua o antigo cluster ou grupo de replicação Valkey ou Redis OSS (modo de cluster desativado). Para obter mais informações, consulte [Excluindo um cluster no ElastiCache](#) ou [Exclusão de um grupo de replicação](#).

4. Dimensione o novo cluster ou grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) para o número de fragmentos (grupos de nós) necessários. Para obter mais informações, consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)
- 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 únicos e OSS clusters com o Multi-AZ desativado, recomendamos que seja disponibilizada memória suficiente para o Redis, OSS conforme descrito em. [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#) Nesses casos, o primário não está disponível para solicitações de serviço durante o processo de atualização.
  - Para OSS clusters Redis com o Multi-AZ ativado, também recomendamos que você agende atualizações do mecanismo durante períodos de baixo tráfego de gravação de entrada. Ao atualizar para o Redis OSS 5.0.6 ou superior, 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 projetado por você mesmo, 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.

## Como atualizar as versões dos mecanismos

### Valkey e Redis OSS

Com o Valkey e o RedisOSS, você inicia atualizações de versão para seu cluster ou grupo de replicação modificando-o usando o ElastiCache console, o ou o ElastiCache API e especificando uma versão mais recente do mecanismo. AWS CLI Para obter mais informações, consulte os tópicos a seguir.

| Como modificar clusters e grupos de replicação              |                                                 |
|-------------------------------------------------------------|-------------------------------------------------|
| Clusters                                                    | Grupos de replicação                            |
| <a href="#">Usando o ElastiCache AWS Management Console</a> | <a href="#">Usando o AWS Management Console</a> |
| <a href="#">Usando o AWS CLI com ElastiCache</a>            | <a href="#">Usando o AWS CLI</a>                |
| <a href="#">Usando o ElastiCache API</a>                    | <a href="#">Usando o ElastiCache API</a>        |

### Memcached

Com o Memcached, para iniciar os upgrades de versão do seu cluster, você o modifica e especifica uma versão mais recente do mecanismo. Você pode fazer isso usando o ElastiCache console AWS CLI, o ou o ElastiCache API:

- Para usar o AWS Management Console, veja —[Usando o ElastiCache AWS Management Console](#).
- Para usar o AWS CLI, consulte [Usando o AWS CLI com ElastiCache](#).
- Para usar o ElastiCache API, consulte [Usando o ElastiCache API](#).

# Como acionar a atualização entre mecanismos do Redis OSS para o Valkey

Você pode atualizar um grupo de OSS replicação existente do Redis (v4 ou posterior) para o mecanismo Valkey usando o Console ou API CLI

## Note

Se você quiser atualizar um cluster de nó único existente do Redis OSS (modo de cluster desativado) para o mecanismo Valkey, você deve primeiro adicioná-lo a um grupo de replicação seguindo estas etapas de pré-requisito: [Criação de um grupo de replicação usando um cluster existente](#)

Se você tiver um grupo de OSS replicação Redis existente que esteja usando o grupo de parâmetros de cache padrão, você pode atualizar para o Valkey especificando o novo mecanismo e a versão do mecanismo com `modify-replication-group` API

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --engine valkey \  
  --engine-version 7.2
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --engine valkey ^  
  --engine-version 7.2
```

Se você tiver um grupo de parâmetros de cache personalizado aplicado ao grupo de replicação redis existente que deseja atualizar, também precisará passar um grupo de parâmetros de cache Valkey personalizado na solicitação. O grupo de parâmetros personalizados Valkey de entrada deve ter os mesmos valores de parâmetros estáticos do Redis que o grupo de parâmetros personalizados do Redis existente.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --engine valkey \  
  --engine-version 7.2 \  
  --cache-parameter-group-name myParamGroup
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --engine valkey ^  
  --engine-version 7.2 ^  
  --cache-parameter-group-name myParamGroup
```

## Atualização entre mecanismos do Redis OSS para o Valkey para Serverless ElastiCache

Você pode atualizar um cache OSS sem servidor do Redis existente para o mecanismo Valkey usando o console API ou CLI especificando o novo mecanismo e a versão principal do mecanismo com o `modify-serverless-cache` API

Para Linux, macOS ou Unix:

```
aws elasticache modify-serverless-cache \  
  --serverless-cache-name myCluster \  
  --engine valkey \  
  --major-engine-version 7
```

Para Windows:

```
aws elasticache modify-serverless-cache ^  
  --serverless-cache-name myCluster ^  
  --engine valkey ^  
  --major-engine-version 7
```

## Mecanismos e versões compatíveis

ElastiCache Os caches sem servidor oferecem suporte ao Valkey 7.2 e superior, ao Redis OSS versão 7.0 e ao Memcached 1.6 e superior.

ElastiCache caches autoprojeto oferecem suporte ao Valkey 7.2 e superior, a todas as OSS versões 4.0.10 e superiores do Redis e às versões 1.4.5 e superiores do Memcached.

ElastiCache Os clusters autoprojeto oferecem suporte às seguintes versões do Valkey:

- [Versões suportadas do Valkey](#)
- [Versões do Redis OSS suportadas](#)
- [Cronograma de fim de vida das OSS versões do Redis](#)
- [Versões suportadas ElastiCache \(Memcached\)](#)

## Versões suportadas do Valkey

Versões suportadas do Valkey abaixo. Observe que o Valkey oferece suporte à maioria dos recursos disponíveis no Redis OSS 7.2 por padrão.

### ElastiCache (Valkey) versão 7.2.6

Em 10 de outubro de 2024, ElastiCache com o Valkey, foi lançado o 7.2.6. Aqui estão alguns dos novos recursos introduzidos no Valkey 7.2 (em comparação com o Redis OSS 7.1):

- Nova WITHSCORE opção para ZREVRANK comandos ZRANK e
- CLIENTNÃO- TOUCH para que os clientes executem comandos sem afetarLRU/LFU das 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.
- Otimizações de desempenho e memória para vários tipos de dados.

Aqui estão as mudanças de comportamento potencialmente significativas entre o Valkey 7.2 e o Redis OSS 7.1 (ou 7.0):

- Ao ligar PUBLISH para um RESP3 cliente que também está inscrito no mesmo canal, o pedido é alterado e a resposta é enviada antes da mensagem publicada.
- 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.
- A amostragem de tempo de congelamento ocorre durante a execução do comando e nos scripts.
- Quando um comando bloqueado está sendo desbloqueado, verificações como ACLOOM, e outras são reavaliadas.
- ACL o texto da mensagem de erro de falha e os códigos de erro são unificados.

- 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).
- As estatísticas do comando são atualizadas para comandos bloqueados somente quando o comando realmente é executado.
- O armazenamento interno dos ACL usuários não remove mais as regras redundantes de comando e categoria. Isso pode alterar a forma como essas regras são exibidas como parte de ACL SAVE ACL GETUSER ACL LIST e.
- Qualquer conexão de cliente criada para uso TLS baseado em replicação, SNI se possível.
- XINFOSTREAM: o campo de resposta de sete tempos agora indica a última tentativa de interação em vez da última interação bem-sucedida. O novo campo de resposta em tempo ativo agora indica a última interação bem-sucedida.
- XREADGROUP e X [AUTO] CLAIM criam o consumidor independentemente de ele ter sido capaz de realizar alguma leitura/reclamação. [TBD- o que é o “isso” aqui?]
- ACLo usuário padrão recém-criado define o sinalizador sanitize-payload em/. ACL LIST GETUSER
- O HELLO comando não afeta o estado do cliente, a menos que seja bem-sucedido.
- NANas respostas são normalizadas para um único tipo nan, semelhante ao comportamento atual de inf.

[Para obter mais informações sobre Valkey, consulte Valkey](#)

[Para obter mais informações sobre a versão 7.2 do Valkey, consulte as notas de lançamento do Redis OSS 7.2.4 \(o Valkey 7.2 inclui todas as alterações do Redis OSS até a versão 7.2.4\) e as notas de lançamento do Valkey 7.2 em Valkey on. GitHub](#)

## Versões do Redis OSS suportadas

ElastiCache Caches sem servidor e caches de design próprio oferecem suporte a todas as versões 7.1 e anteriores do RedisOSS.

- [ElastiCache \(Valkey\) versão 7.2.6](#)
- [ElastiCache \(RedisOSS\) versão 7.1 \(aprimorada\)](#)

ElastiCache Os clusters autoprotetidos oferecem suporte às seguintes versões do Valkey e do RedisOSS:

- [ElastiCache \(RedisOSS\) versão 7.1 \(aprimorada\)](#)

- [ElastiCache \(RedisOSS\) versão 7.0 \(aprimorada\)](#)
- [ElastiCache \(RedisOSS\) versão 6.2 \(aprimorada\)](#)
- [ElastiCache \(RedisOSS\) versão 6.0 \(aprimorada\)](#)
- [ElastiCache \(RedisOSS\) versão 5.0.6 \(aprimorada\)](#)
- [ElastiCache \(RedisOSS\) versão 5.0.5 \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versão 5.0.4 \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versão 5.0.3 \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versão 5.0.0 \(obsoleta, use a versão 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versão 4.0.10 \(aprimorada\)](#)
- [Versões do Past End of Life \(EOL\) \(3.x\)](#)
- [Versões do Past End of Life \(EOL\) \(2.x\)](#)

## ElastiCache (RedisOSS) versão 7.1 (aprimorada)

Esta versão contém melhorias de desempenho que permitem que as cargas de trabalho aumentem a produtividade e reduzam as latências de operação. ElastiCache 7.1 apresenta [dois aprimoramentos principais](#):

Estendemos a funcionalidade de threads de E/S melhorada para também lidar com a lógica da camada de apresentação. Por camada de apresentação, queremos dizer os threads de E/S aprimorados que agora não estão apenas lendo a entrada do cliente, mas também analisando a entrada no formato de comando OSS binário do Redis. Em seguida, isso é encaminhado para o thread principal para execução, o que proporciona um ganho de desempenho. Padrão de acesso à OSS memória Redis 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 ElastiCache (RedisOSS) v7.1, os clientes podem obter até 100% mais taxa de transferência e 50% menos latência P99 em relação ao (Redis) v7.0. ElastiCache OSS Esses aprimoramentos são habilitados em tamanhos de nós com pelo menos 8 núcleos físicos (2xlarge Graviton e 4xlarge no x86), independentemente do CPU tipo, e não exigem alterações no cliente.

### Note

ElastiCache A v7.1 é compatível com o OSS Redis v7.0.



## ElastiCache (RedisOSS) versão 7.0 (aprimorada)

ElastiCache (RedisOSS) 7.0 adiciona uma série de melhorias e suporte para novas funcionalidades:

- **Funções:** ElastiCache (RedisOSS) 7 adiciona suporte às OSS Funções do Redis e fornece uma experiência gerenciada que permite que os desenvolvedores executem [LUAScripts](#) com a lógica do aplicativo armazenada no ElastiCache cluster, sem exigir que os clientes reenviem os scripts para o servidor a cada conexão.
- **ACLmelhorias:** Valkey e Redis OSS 7 adicionam suporte para a próxima versão do Access Control Lists (ACLs). Agora, os clientes podem especificar vários conjuntos de permissões em chaves ou espaços de chave específicos no Valkey e no Redis. OSS
- **Pub/Sub fragmentado:** ElastiCache com Valkey e Redis, o OSS 7 adiciona suporte para executar a funcionalidade do Pub/Sub de forma fragmentada ao executar no modo de cluster ativado (CME). Os recursos do Pub/Sub 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.
- **Multiplexação de E/S aprimorada:** ElastiCache com 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. 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. ElastiCache OSS

Para obter mais informações sobre o Valkey, consulte [Valkey](#). Para obter mais informações sobre a versão do Redis OSS 7.0, consulte as [notas de lançamento do Redis OSS 7.0](#) em Redis on. OSS GitHub

## ElastiCache (RedisOSS) versão 6.2 (aprimorada)

ElastiCache (RedisOSS) 6.2 inclui melhorias de desempenho para clusters TLS habilitados usando tipos de nós x86 com 8 vCPUs ou mais ou tipos de nós Graviton2 com 4 ou mais vCPUs. Esses aprimoramentos melhoram a taxa de transferência e reduzem o tempo de estabelecimento da conexão do cliente ao transferir a criptografia para outros vCPUs. 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 classificação de dados em camadas em nós de cluster contendo NVMe SSD anexos locais. Para obter mais informações, consulte [Hierarquização de dados em ElastiCache](#).

A versão 6.2.6 OSS do mecanismo Redis também introduz suporte ao formato nativo JavaScript Object Notation (JSON), uma forma simples e sem esquemas de codificar conjuntos de dados complexos dentro de clusters do Redis. OSS Com o JSON suporte, você pode aproveitar o desempenho e o Redis OSS APIs para aplicativos que operam em excessoJSON. Para obter mais informações, consulte [Introdução ao JSON](#). Também estão incluídas métricas JSON relacionadas `JsonBasedCmds` e `JsonBasedCmdsLatency` incorporadas CloudWatch para monitorar o uso desse tipo de dados. Para obter mais informações, consulte [Métricas para Valkey e Redis OSS](#).

Você especifica a versão do mecanismo usando 6.2. ElastiCache (RedisOSS) invocará automaticamente a versão de patch preferida do Redis OSS 6.2 que está disponível. Por exemplo, ao criar/modificar um cluster de cache, 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 da criação/modificação. Especificar a versão 6.x do motor no API resultará na última versão secundária 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 (RedisOSS) 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 o `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 OSS 6.2 do Redis, consulte as [notas de lançamento do Redis OSS 6.2](#) em Redis on. OSS GitHub

ElastiCache (RedisOSS) versão 6.0 (aprimorada)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão do OSS mecanismo Redis, que inclui a [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 (RedisOSS) oferecerá uma única versão para cada versão OSS secundária do Redis, em vez de oferecer várias versões de patch. ElastiCache (RedisOSS) gerenciará automaticamente a versão de patch de seus clusters de cache 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 (RedisOSS) 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 (RedisOSS) invocará automaticamente a versão de patch preferida do Redis OSS 6.0 que está disponível. Por exemplo, ao criar/modificar um cluster de cache, você define o parâmetro `--engine-version` como `6.0`. O cluster será lançado com a versão de patch preferencial disponível atualmente 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 o `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 OSS 6.0 do Redis, consulte as [notas de lançamento do Redis OSS 6.0](#) em Redis on. OSS GitHub

ElastiCache (RedisOSS) versão 5.0.6 (aprimorada)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão do OSS mecanismo Redis, 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 chamado `rename-commands` assim permite renomear OSS comandos do Redis potencialmente perigosos ou caros que podem causar perda acidental de dados, como `ou. FLUSHALL FLUSHDB` Isso é semelhante à configuração do comando

renomear no Redis de código aberto. OSS 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 contém 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 nos parâmetros do Redis OSS 5.0.3](#).

- Redis OSS Streams: isso modela uma estrutura de dados de log que permite aos produtores acrescentar 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 [Streams](#).
- Oferece suporte para uma família de comandos de fluxo, como os XADD, XRANGE e XREAD. Para obter mais informações, consulte [Streams Commands](#).
- Diversos parâmetros novos e renomeados. Para obter mais informações, consulte [Alterações nos parâmetros do Redis OSS 5.0.0](#).
- Uma nova OSS métrica do Redis, `StreamBasedCmds`.
- Tempo de captura de imagem um pouco mais rápido para os nós do RedisOSS.

#### Important

ElastiCache (RedisOSS) retrocedeu duas correções de bugs críticos da versão 5.0.1 de [OSS código aberto do Redis](#). Eles estão listados a seguir:

- `RESTORE` 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 ElastiCache (RedisOSS) para a versão 5.0.0 OSS do mecanismo Redis 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 (RedisOSS) versão 5.0.5 (obsoleta, use a versão 5.0.6)

Amazon ElastiCache (RedisOSS) apresenta a próxima versão do mecanismo RedisOSS;. Ele inclui alterações de configuração on-line para ElastiCache (RedisOSS) de clusters de failover automático durante todas as operações planejadas. Agora você pode escalar seu cluster, atualizar a versão do OSS mecanismo Redis e aplicar patches e atualizações de manutenção enquanto o cluster permanece on-line 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 (RedisOSS) versão 5.0.4 (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão do OSS mecanismo Redis suportada pela Amazon. 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 (RedisOSS) versão 5.0.3 (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão do OSS mecanismo Redis suportada pela Amazon, ElastiCache que inclui correções de bugs.

## ElastiCache (RedisOSS) versão 5.0.0 (obsoleta, use a versão 5.0.6)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão principal do OSS mecanismo Redis suportada pela Amazon. ElastiCache ElastiCache (RedisOSS) 5.0.0 oferece suporte para as seguintes melhorias:

- Redis OSS Streams: isso modela uma estrutura de dados de log que permite aos produtores acrescentar 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 [Streams](#).
- Oferece suporte para uma família de comandos de fluxo, como os XADD, XRANGE e XREAD. Para obter mais informações, consulte [Streams Commands](#).
- Diversos parâmetros novos e renomeados. Para obter mais informações, consulte [Alterações nos parâmetros do Redis OSS 5.0.0](#).
- Uma nova OSS métrica do Redis, `StreamBasedCmds`.
- Tempo de captura de imagem um pouco mais rápido para os nós do RedisOSS.

## ElastiCache (RedisOSS) versão 4.0.10 (aprimorada)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão principal do OSS mecanismo Redis suportada pela Amazon. ElastiCache ElastiCache (RedisOSS) 4.0.10 oferece suporte às seguintes melhorias:

- Redimensionamento e criptografia de clusters on-line em uma única versão ElastiCache (RedisOSS). Para obter mais informações, consulte as informações a seguir.
  - [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)
  - [Refragmentação online para Valkey ou Redis OSS \(modo de cluster ativado\)](#)
  - [Segurança de dados na Amazon ElastiCache](#)
- Diversos novos parâmetros. Para obter mais informações, consulte [Alterações nos parâmetros do Redis OSS 4.0.10](#).

- Suporte para a família de comandos de memória, como MEMORY. Para obter mais informações, consulte [Comandos](#) (pesquisa ativada MEMO).
- Suporte para desfragmentação de memória online, permitindo assim a utilização mais eficiente da memória e uma quantidade maior de memória disponível para seus dados.
- Suporte para descargas e exclusões assíncronas. ElastiCache (RedisOSS) 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 OSS métrica do Redis, ActiveDefragHits. Para obter mais informações, consulte [Métricas para Redis OSS](#).

Usuários do Valkey ou do Redis OSS (modo de cluster desativado) que executam a OSS versão 3.2.10 do Redis podem usar o console para atualizar seus clusters por meio da atualização on-line.

Comparando o suporte ao redimensionamento e criptografia de clusters ElastiCache OSS (Redis)

| Atributo                              | 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                |

\* Adição, remoção e rebalanceamento de fragmentos.

\*\* Necessário para aplicativos Fed RAMP e PCI DSS compatíveis. HIPAA Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

Versões do Past End of Life (EOL) (3.x)

ElastiCache (RedisOSS) versão 3.2.10 (aprimorada)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão principal do OSS mecanismo Redis suportada pela Amazon. ElastiCache ElastiCache (RedisOSS) 3.2.10 introduz o redimensionamento on-line do cluster para adicionar ou remover fragmentos do cluster enquanto ele continua atendendo às solicitações de E/S recebidas. ElastiCache Os usuários do (RedisOSS) 3.2.10 têm todas as



funcionalidades das OSS versões anteriores do Redis, exceto a capacidade de criptografar seus dados. Esse recurso está atualmente disponível apenas na versão 3.2.6.

Comparando ElastiCache (RedisOSS) as versões 3.2.6 e 3.2.10

| Atributo                              | 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.

\*\* Necessário para aplicativos Fed RAMP e PCI DSS compatíveis. HIPAA Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

Para obter mais informações, consulte as informações a seguir.

- [Refragmentação online para Valkey ou Redis OSS \(modo de cluster ativado\)](#)
- [Redimensionamento de cluster on-line](#)

ElastiCache (RedisOSS) versão 3.2.6 (aprimorada)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão principal do OSS mecanismo Redis suportada pela Amazon. ElastiCache ElastiCache Os usuários do (RedisOSS) 3.2.6 têm todas as funcionalidades das OSS versões anteriores do Redis, além da opção de criptografar seus dados. Para obter mais informações, consulte as informações a seguir.

- [ElastiCache criptografia em trânsito \(\) TLS](#)
- [Criptografia em repouso em ElastiCache](#)
- [Validação de conformidade para a Amazon ElastiCache](#)

ElastiCache (RedisOSS) versão 3.2.4 (aprimorada)

A versão 3.2.4 do Amazon ElastiCache (RedisOSS) apresenta a próxima versão principal do mecanismo Redis OSS suportado pela Amazon. ElastiCache ElastiCache Os usuários do

(RedisOSS) 3.2.4 têm todas as funcionalidades das OSS versões anteriores do Redis disponíveis, além da opção de execução no modo de cluster ou no modo não cluster. A tabela a seguir fornece um resumo.

Comparando o modo sem cluster e o modo cluster do Redis OSS 3.2.4

| Atributo                    | 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          | <code>default.redis3.2</code> ** | <code>default.redis3.2.cluster.on</code> ** |

\* 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 introduz suporte para indexação geoespacial por meio de seis comandos. GEO Para obter mais informações, consulte a documentação do comando Redis OSS GEO \* [Comandos: GEO](#) na página Comandos do Valkey (filtrada por). GEO

Para obter informações sobre recursos adicionais do Redis OSS 3, consulte as notas de lançamento do [Redis OSS 3.2 e as notas de lançamento](#) do [Redis OSS 3.0](#).

O Valkey ou o Redis atualmente ElastiCache gerenciados OSS (modo de cluster ativado) não são compatíveis com os seguintes recursos do Redis 3.2OSS:

- 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 nos parâmetros do Redis OSS 3.2.4](#)

Versões do Past End of Life (EOL) (2.x)

ElastiCache (RedisOSS) versão 2.8.24 (aprimorada)

OSSAs melhorias do Redis adicionadas desde a versão 2.8.23 incluem correções de bugs e registro de endereços de acesso incorretos à memória. Para obter mais informações, consulte as [notas de versão do Redis OSS 2.8](#).

## ElastiCache (RedisOSS) versão 2.8.23 (aprimorada)

OSSAs melhorias do Redis adicionadas desde a versão 2.8.22 incluem correções de bugs. Para obter mais informações, consulte as [notas de versão 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 o Guia do [Parâmetros adicionados ao Redis OSS 2.8.23 \(aprimorado\)](#) ElastiCache usuário.

## ElastiCache (RedisOSS) versão 2.8.22 (aprimorada)

OSSAs melhorias do Redis 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](#).

- Várias correções de erros críticos no PSYNC comportamento de replicação. Para obter mais informações, consulte as [notas de versão 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.

- Variáveis de OSS configuração do Redis `appendonly` e não `appendfsync` são compatíveis com a OSS versão 2.8.22 e posterior do Redis.
- 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. É mais provável que tais situações ocorram para PUBSUB os clientes.

#### ElastiCache (RedisOSS) versão 2.8.21

OSSAs melhorias do Redis adicionadas desde a versão 2.8.19 incluem várias correções de bugs. Para obter mais informações, consulte as [notas de versão do Redis OSS 2.8](#).

#### ElastiCache (RedisOSS) versão 2.8.19

OSSAs melhorias do Redis adicionadas desde a versão 2.8.6 incluem o seguinte:

- Support for HyperLogLog. Para obter mais informações, consulte a [OSSnova estrutura de dados do Redis: 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 os nós de réplica, o mestre SYNC falhará se um processo secundário `save` (`bgsave`) em segundo plano for abortado.
- Support para a `HyperLogLogBasedCommands` CloudWatch métrica. Para obter mais informações, consulte [Métricas para Valkey e Redis OSS](#).

#### ElastiCache (RedisOSS) versão 2.8.6

OSSAs melhorias do Redis adicionadas desde a versão 2.6.13 incluem o seguinte:

- Resistência aprimorada e tolerância a falhas para réplicas de leitura.
- Suporte para resincronização 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 (RedisOSS) versão 2.6.13

A OSS versão 2.6.13 do Redis foi a versão inicial do Redis suportada OSS pela Amazon ElastiCache (Redis). OSS O Multi-AZ não é compatível com o Redis OSS 2.6.13.

### Cronograma de fim de vida das OSS versões do Redis

Esta seção define as datas de fim da vida útil (EOL) para as versões principais mais antigas à medida que elas são anunciadas. Ela ajudará você a tomar decisões sobre versões e atualizações no futuro.


#### Note

ElastiCache As versões de patch (RedisOSS) de 5.0.0 a 5.0.5 estão obsoletas. Use as versões 5.0.6 ou superior.


A tabela a seguir resume cada versão e sua EOL data de anúncio, bem como a versão alvo de atualização recomendada.

#### Passado EOL

| Versões secundárias de origem | Destino de atualização recomendado | EOLData             |
|-------------------------------|------------------------------------|---------------------|
| 3.2.4, 3.2.6 e 3.2.10         | Versão 6.2 ou superior             | 31 de julho de 2023 |

 **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.

| Versões secundárias de origem                                    | Destino de atualização recomendado                                                                                                                                                                                                                                                                                                                                                            | EOLData               |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| 2.8.24, 2.8.23, 2.8.22, 2.8.21,<br>2.8.19, 2.8.12, 2.8.6, 2.6.13 | Versão 6.2 ou superior<br><br><div data-bbox="613 447 1040 898" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b></p> <p>Para as regiões US-ISO-EAST US-ISO-WEST -1, -1 e US-ISOB-EAST -1, recomendamos a versão 5.0.6 ou superior.</p> </div> | 13 de janeiro de 2023 |

## Versões suportadas ElastiCache (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 às condições que, caso não sejam atendidas, farão com que a atualização falhe.

### ElastiCache para versões do Memcached

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

- [Memcached versão 1.4.5](#)

#### Memcached versão 1.6.22

ElastiCache (Memcached) adiciona suporte para a versão 1.6.22 do Memcached. 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

#### Memcached versão 1.6.17

ElastiCache (Memcached) adiciona suporte para a versão 1.6.17 do 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

#### Memcached versão 1.6.12

ElastiCache (Memcached) adiciona suporte para o Memcached versão 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

#### Memcached versão 1.6.6

ElastiCache (Memcached) adiciona suporte para a versão 1.6.6 do Memcached. Ele não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.5.16](#). ElastiCache [\(Memcached\) não inclui suporte para Extstore](#).

Para obter mais informações, consulte [ReleaseNotes166](#) em Memcached on. GitHub

#### Memcached versão 1.5.16

ElastiCache for Memcached adiciona suporte para Memcached versão 1.5.16. 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

#### Memcached versão 1.5.10

ElastiCache para Memcached, a versão 1.5.10 oferece suporte aos seguintes recursos do Memcached:



- Rebalanceamento de slab automatizado.
- Pesquisas de tabela de hash mais rápidas com o algoritmo murmur3.
- LRUAlgoritmo segmentado.
- LRURastreador para recuperar memória em segundo plano.
- `--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 versão ElastiCache 1.4.34 do Memcached incluem o seguinte:

- Correções cumulativas, como ASCII multiget, CVE -2017-9951 e rastreamento de limite para `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

#### Memcached versão 1.4.34

ElastiCache para Memcached, a versão 1.4.34 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

#### Memcached versão 1.4.33

As melhorias no Memcached adicionadas desde a versão 1.4.24 incluem:

- 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](#).

#### Memcached versão 1.4.24

As melhorias no Memcached adicionadas desde a versão 1.4.14 incluem:

- Gerenciamento usado menos recentemente (LRU) 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.

#### Memcached versão 1.4.14

As melhorias no Memcached adicionadas desde a versão 1.4.5 incluem:

- 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.

## Memcached versão 1.4.5

A versão 1.4.5 do Memcached foi o mecanismo inicial e a versão suportados pela Amazon ElastiCache (Memcached).

## Principais diferenças de comportamento e compatibilidade da versão com o Valkey

O Valkey 7.2.6 tem diferenças de compatibilidade semelhantes às versões anteriores do Redis OSS 7.2.5. Para obter a versão mais recente compatível do Valkey, consulte [Mecanismos e versões compatíveis](#).

## Principais diferenças de comportamento e compatibilidade 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 atualizar diretamente da sua versão atual do Redis para a OSS versão mais recente disponível do RedisOSS, sem a necessidade de atualizações sequenciais. Por exemplo, você pode atualizar diretamente da OSS versão 3.0 do Redis para a versão 7.0.

OSSAs versões do Redis são identificadas com uma versão semântica que compreende um componente MAJORMINOR, e. PATCH Por exemplo, no Redis OSS 4.0.10, a versão principal é 4, a versão secundária 0 e a versão do patch é 10. Esses valores geralmente são incrementados com base nas seguintes convenções:

- MAJORas versões são para API alterações incompatíveis
- MINORas versões são para novas funcionalidades adicionadas de forma compatível com versões anteriores
- PATCHas versões são para correções de bugs compatíveis com versões anteriores e alterações não funcionais

Recomendamos sempre manter a versão mais recente do patch em um determinado momento MAJOR. MINOR versão para ter as últimas melhorias de desempenho e estabilidade. A partir do Redis OSS 6.0, ElastiCache (RedisOSS) oferecerá uma única versão para cada versão OSS secundária do Redis, em vez de oferecer várias versões de patch. ElastiCache (RedisOSS)

gerenciará automaticamente a versão de patch de seus clusters de cache 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 ElastiCache (RedisOSS) oferece suporte às duas mais recentes. MAJOR MINOR versões da época para a nova região. Por exemplo, se uma nova AWS região for lançada e a mais recente MAJOR. MINOR ElastiCache As versões (RedisOSS) são 7.0 e 6.2, ElastiCache (RedisOSS) suportará as versões 7.0 e 6.2 na nova região. AWS Como mais novo MAJOR. MINOR versões do ElastiCache (RedisOSS) foram lançadas e ElastiCache continuarão adicionando suporte às versões recém-lançadas ElastiCache (RedisOSS). Para saber mais sobre como escolher regiões para ElastiCache, consulte [Escolha de regiões e zonas de disponibilidade](#).

Ao fazer uma atualização que abranja versões principais ou secundárias, considere a lista a seguir, que inclui alterações comportamentais e incompatíveis com versões anteriores lançadas com o Redis OSS ao longo do tempo.

## Comportamento do Redis OSS 7.0 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, consulte as notas de [lançamento do Redis OSS 7.0](#).

- `SCRIPT LOAD` e `SCRIPT FLUSH` não são mais propagados para réplicas. Se você precisar de alguma durabilidade para scripts, recomendamos que considere usar as [OSS funções do Redis](#).
- Os canais do Pubsub agora estão bloqueados por padrão para novos usuários. 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 ACL categorias 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.

## Comportamento do Redis OSS 6.2 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, consulte as notas de [lançamento do Redis OSS 6.2](#).

- As ACL bandeiras dos LASTSAVE comandos TIME, ECHOROLE, e foram alteradas. 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 pares chave/valor retornados de uma resposta do mapa a um script lua é alterada. Se seus scripts usam `redis.setresp()` ou retornam um mapa (novo no Redis OSS 6.0), considere as implicações de que o script pode falhar nas atualizações.

## Comportamento do Redis OSS 6.0 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, consulte as notas de [lançamento 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 (RedisOSS) 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](#).

## Comportamento do Redis OSS 5.0 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, consulte as [notas de lançamento 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 no ElastiCache (RedisOSS) 5.0.

- Se você estiver atualizando do Redis OSS 4.0, alguns comandos nos LUA scripts retornarão argumentos em uma ordem diferente da das versões anteriores. No Redis OSS 4.0, o Redis OSS ordenaria algumas respostas lexicograficamente para tornar as respostas determinísticas. Essa ordenação não é aplicada quando os scripts são replicados por efeitos.
- No Redis OSS 5.0.3 e superior, ElastiCache (RedisOSS) 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 desempenho 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.

## Comportamento do Redis OSS 4.0 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, 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.

## Passado EOL

### Comportamento do Redis OSS 3.2 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, consulte as [notas de lançamento do Redis OSS 3.2](#).

- Não há mudanças de compatibilidade a serem destacadas para esta versão.

Para obter mais informações, consulte [Cronograma de fim de vida das OSS versões do Redis](#).

### Comportamento do Redis OSS 2.8 e alterações incompatíveis com versões anteriores

Para ver uma lista completa das mudanças, consulte as [notas de lançamento do Redis OSS 2.8](#).

- A partir do Redis OSS 2.8.22, o Redis não OSS AOF é mais suportado no ElastiCache (Redis). OSS Recomendamos o uso do MemoryDB quando os dados precisam ser persistidos de forma duradoura.
- A partir do Redis OSS 2.8.22, ElastiCache (RedisOSS) não dá mais suporte à 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 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 [Cronograma de fim de vida das OSS versões do Redis](#).

## Resolvendo atualizações bloqueadas do motor Valkey ou Redis OSS

Conforme mostrado na tabela a seguir, sua operação de atualização do OSS mecanismo Valkey ou Redis será bloqueada se você tiver uma operação de aumento de escala pendente.

| 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 OSS do mecanismo Redis

- Execute um destes procedimentos:
  - Agende sua operação de atualização OSS do mecanismo Redis para a próxima janela de manutenção desmarcando a caixa de seleção Aplicar imediatamente.



Com o CLI, use `--no-apply-immediately`. Com o API, use `useApplyImmediately=false`.

- Espere até a próxima janela de manutenção (ou depois) para realizar a operação de atualização OSS do mecanismo Redis.
- Adicione a operação de OSS escalonamento do Redis a essa modificação do cluster com a caixa de seleção Aplicar imediatamente selecionada.

Com o CLI, use `--apply-immediately`. Com o API, use `useApplyImmediately=true`.

Essa abordagem cancela efetivamente a atualização do mecanismo durante a próxima janela de manutenção, executando-a imediatamente.

## ElastiCache melhores práticas e estratégias de armazenamento em cache

Abaixo, você encontra as melhores práticas recomendadas para a Amazon ElastiCache. Seguir essas práticas melhora o desempenho e aumenta a confiabilidade do cache.

### Tópicos

- [Melhores práticas gerais](#)
- [Comandos Valkey, Redis e Memcached suportados OSS e restritos](#)
- [OSSConfiguração e limites do Valkey e do Redis](#)
- [IPv6exemplos de clientes para Valkey, Redis OSS e Memcached](#)
- [Melhores práticas para clientes \(Valkey e RedisOSS\)](#)
- [Melhores práticas para clientes \(Memcached\)](#)
- [TLSclusters de pilha ElastiCache dupla habilitados](#)
- [Gerenciando a memória reservada para Valkey e Redis OSS](#)
- [Práticas recomendadas ao trabalhar com clusters autoprojeados do Valkey e do Redis OSS](#)
- [Estratégias de armazenamento em cache para Memcached](#)

## Melhores práticas gerais

Abaixo, você encontra informações sobre as melhores práticas para usar as interfaces Valkey, Redis OSS e Memcached. 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.
- Use conexões de longa duração — Criar uma nova conexão é caro e consome tempo e CPU recursos 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 tiver réplicas de leitura provisionadas (clusters autoprojeto), direcione as leituras para as réplicas para obter melhor escalabilidade e/ou menor latência. As leituras de réplicas acabarão sendo consistentes com a primária.

Em um cluster autoprojeto, 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 com uso intensivo computacional e de E/S, 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 cargas de trabalho pub/sub com alta taxa de transferência, recomendamos que você use o pub/sub [fragmentado \(disponível com o Valkey e com o Redis 7 ou posterior\)](#). OSS O pub/sub tradicional em clusters habilitados para modo cluster transmite mensagens para todos os nós do cluster, o que pode acarretar EngineCPUUtilization elevada. Observe que, ElastiCache sem servidor, os comandos pub/sub tradicionais usam internamente comandos pub/sub fragmentados.

# Comandos Valkey, Redis e Memcached suportados OSS e restritos

## Comandos Valkey e Redis OSS suportados

### Comandos Valkey e Redis OSS suportados

Os seguintes OSS comandos do Valkey e do Redis são compatíveis com caches sem servidor. Além desses comandos, estes [Comandos Valkey e Redis OSS suportados](#) também são compatíveis.

#### 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\_RO

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](#)

- READONLY

Permite consultas somente de leitura para uma conexão com um nó de réplica do Valkey ou do Redis OSS Cluster.

[Saiba mais](#)

- READWRITE

Permite consultas de leitura e gravação para uma conexão com um nó de réplica do Valkey ou do Redis OSS Cluster.

[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

Apertos de mão com o servidor 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.

- **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

Extrai 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

Extrai 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](#)

- BLPOP

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

Extrai 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](#)

- RPUISH

Acrescenta um ou mais elementos a uma lista. Cria a chave, caso ela não exista.


[Saiba mais](#)

- RPUISHX

Acrescenta um elemento a uma lista somente quando a lista existe.

[Saiba mais](#)

## Comandos Pub/Sub

 Note

PUBSUBOs comandos usam fragmentados internamentePUBSUB, então os nomes dos 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.

[PUBSUB-SHARDCHANNELS](#)

- PUBSUB SHARDNUMSUB

Retorna a contagem de assinantes de canais de fragmento.

[PUBSUB-SHARDNUMSUB](#)

- 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

- ACL CAT

Lista as ACL categorias ou os comandos dentro de uma categoria.

[Saiba mais](#)

- ACL GENPASS

Gera uma senha pseudo-aleatória e segura que pode ser usada para identificar usuários. ACL

[Saiba mais](#)

- ACL GETUSER

Lista as ACL regras de um usuário.

[Saiba mais](#)

- ACL LIST

Coloca as regras efetivas em formato de ACL arquivo.

[Saiba mais](#)

- ACL USERS

Lista todos os ACL usuários.

[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](#)

- **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 de computador e a versão Valkey ou RedisOSS.

[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](#)

- SCARDT

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.

- 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 atonicamente os valores de string de uma ou mais chaves.

[Saiba mais](#)

- MSET

Cria ou modifica atonicamente os valores de string de uma ou mais chaves.

[Saiba mais](#)

- MSETNX

Só modifica atonicamente 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, 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
- fcall
- fcall\_ro
- function
- function delete
- function dump
- function flush
- function help
- function kill
- function list

- function load
- function restore
- function stats
- keys
- lastsave
- latency
- latency doctor
- latency graph
- latency help
- latency histogram
- latency history
- latency latest
- latency reset
- memory
- memory doctor
- memory help
- memory malloc-stats
- memory purge
- memory stats
- memory usage
- monitor
- move
- object
- object encoding
- object freq
- object help
- object idletime
- object refcount
- pfdebug
- pfselftest

- `psubscribe`
- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`
- `wait`
- `watch`

## 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:

- As conexões do cliente são necessárias TLS, como resultado, o UDP protocolo não é suportado.
- 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 do LRU e do LRU rastreador
  - `watch`: observa os logs de servidor do memcached
  - `verbosity`: configura o nível de log do servidor
  - `me`- o comando meta debug (`me`) não é suportado

## OSSConfiguração e limites do Valkey e do Redis

Cada um dos OSS mecanismos Valkey e Redis fornece vários parâmetros de configuração, alguns dos quais são modificáveis em ElastiCache (RedisOSS) e outros não são modificáveis para fornecer desempenho e confiabilidade estáveis.

### Caches sem servidor

Para caches sem servidor, grupos de parâmetros não são usados e todas as configurações do Valkey ou do Redis OSS não são modificáveis. Os seguintes OSS parâmetros do Valkey ou do Redis estão em vigor:

| Nome                                | Detalhes                           | Descrição                                                                                                                                                |
|-------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| acl-pubsub-default                  | allchannels                        | Permissões padrão do canal pubsub para ACL usuários no cache.                                                                                            |
| client-output-buffer-limit          | normal 0 0 0<br>pubsub 32mb 8mb 60 | Cientes normais não têm limite de buffer. PUB/SUBos clientes serão desconectados se violarem o backlog de 32 MiB ou violarem o backlog de 8 MiB por 60s. |
| client-query-buffer-limit           | 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 4.000 argumentos.       |
| cluster-allow-pubsubshard-when-down | yes                                | Isso permite que o cache ofereça tráfego pubsub enquanto o cache está parcialmente inativo.                                                              |
| cluster-allow-reads-when-down       | yes                                | Isso permite que o cache ofereça tráfego de leitura enquanto o cache está parcialmente inativo.                                                          |



| Nome                                       | Detalhes | Descrição                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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.                                                                                                                                                                                                                                                                                 |
| <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     | <p>O tempo máximo de execução de um script Lua, em milissegundos, antes de ElastiCache agir para interromper o script.</p> <p>Se <i>lua-time-limit</i> for excedido, todos os OSS comandos Valkey ou Redis podem retornar um erro no formato <code>____-BUSY</code>. Como esse estado pode causar interferência em muitas OSS operações essenciais do Valkey ou do Redis, primeiro ElastiCache emitirá um <code>SCRIPTKILL</code> comando. Se isso não for bem-sucedido, ElastiCache reiniciará forçosamente o Valkey ou o Redis. OSS</p> |
| <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.                                                                                                                                                                                                                                                                                                                                                                                                 |

| Nome                   | Detalhes                                       | Descrição                                                                                                                                                                                                                        |
|------------------------|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| maxmemory-policy       | volatile-lru                                   | Os itens com um TTL conjunto são despejados seguindo a estimativa least-recently-used (LRU) quando o limite de memória do cache é atingido.                                                                                      |
| notify-keyspace-events | (uma string vazia)                             | No momento, os eventos do espaço de chaves não são compatíveis com caches sem servidor.                                                                                                                                          |
| port                   | Porta primária: 6379<br>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 | Descrição                                                                                                                                                    |
|--------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tamanho do nome da chave | 8 KiB    | O tamanho máximo de uma única chave Valkey ou Redis ou OSS nome de canal. Os clientes que fizerem referência a chaves maiores do que isso receberão um erro. |

| Nome                  | Detalhes | Descrição                                                                                                                                                                                                                                                                            |
|-----------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tamanho do script Lua | 50 MiB   | O tamanho máximo de um único script Valkey ou Redis OSS Lua. As tentativas de carregar um script Lua maior do que isso receberão um erro.                                                                                                                                            |
| Tamanho do slot       | 32 GiB   | O tamanho máximo de um único slot de OSS hash Valkey ou Redis. Os clientes que tentarem definir mais dados do que isso em um único OSS slot Valkey ou Redis acionarão a política de despejo no slot e, se nenhuma chave for removível, receberão um erro de falta de memória (). OOM |

## Clusters autoprotetados

Para clusters autoprotetados, consulte [Parâmetros Valkey e 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.

## IPv6exemplos de clientes para Valkey, Redis OSS e Memcached

ElastiCache é compatível com Valkey, Redis OSS e Memcached. Isso significa que os clientes que oferecem suporte a IPv6 conexões devem poder se conectar a clusters IPv6 habilitados ElastiCache (Memcached). Há algumas ressalvas que vale a pena observar ao interagir com IPv6 recursos habilitados.

Você pode ver a postagem do blog de [melhores práticas para clientes Valkey e Redis](#) no blog do AWS banco de dados para obter recomendações sobre como configurar clientes Valkey e OSS Redis para obter recursos. ElastiCache

A seguir estão as melhores práticas para interagir com ElastiCache recursos IPv6 habilitados com bibliotecas cliente de código aberto comumente usadas.

## Cientes validados com Valkey e Redis OSS

ElastiCache é compatível com Valkey e Redis de código aberto. Isso significa que os OSS clientes Valkey e Redis de código aberto que oferecem suporte a IPv6 conexões devem ser capazes de se conectar a clusters IPv6 habilitados ElastiCache (RedisOSS). 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 suportadas (IPv4somente, IPv6 somente e Dual Stack)

Os seguintes clientes foram validados especificamente para funcionar com todas as configurações de tipo de rede suportadas para Valkey e Redis. OSS

Cientes validados:

- [Redis Py \(\) – 4.1.2](#)
- [Lettuce — Versão: 6.1.6. RELEASE](#)
- [Jedis – Versão: 3.6.0](#)

## Melhores práticas para clientes (Valkey e RedisOSS)

Conheça as melhores práticas para cenários comuns e acompanhe exemplos de código de algumas das bibliotecas OSS cliente Valkey e Redis de código aberto mais populares (redis-py e Lettuce)PHPRedis, bem como as melhores práticas para interagir com ElastiCache recursos com bibliotecas cliente de código aberto do Memcached comumente usadas.

Tópicos

- [Grande número de conexões \(Valkey e RedisOSS\)](#)
- [Detecção de clientes em 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 grandes itens compostos \(Valkey e Redis\) OSS](#)
- [Configuração do cliente Lettuce \(Valkey e Redis\) OSS](#)
- [Configurando um protocolo preferencial para clusters de pilha dupla \(Valkey e Redis\) OSS](#)

## Grande número de conexões (Valkey e RedisOSS)

Caches sem servidor e nós individuais ElastiCache (RedisOSS) suportam até 65.000 conexões simultâneas de clientes. 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 têm, OSS cada um, um processo de thread único baseado em um loop de eventos em que as solicitações recebidas do cliente são tratadas sequencialmente. Isso significa que o tempo de resposta de um determinado cliente aumenta à medida que o número de clientes conectados aumenta.

Você pode realizar o seguinte conjunto de ações para evitar um gargalo de conexão em um servidor Valkey ou Redis: OSS

- Realize operações de leitura a partir das réplicas de leitura. Isso pode ser feito usando os endpoints do ElastiCache leitor no modo de cluster desativado ou usando réplicas para leituras no modo de cluster ativado, incluindo um cache sem servidor.
- Distribua tráfego de gravação em vários nós primários. É possível fazer isso de duas maneiras. Você pode usar um OSS cluster Valkey ou Redis com vários fragmentos com um cliente compatível com o modo de 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 TCP conexão é uma operação computacionalmente cara em comparação com os comandos típicos do Valkey ou do Redis. OSS Por exemplo, lidar com uma GET solicitaçãoSET/ 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 exemplo de código a seguir 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 (AmazonEC2) conectada a um nó Graviton2 (m6g.2xlarge) (Redis). ElastiCache 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);  
}
```

}

## Detecção de clientes em cluster e recuo exponencial (Valkey e Redis) OSS

Ao se conectar a um OSS cluster ElastiCache Valkey ou Redis no modo de cluster ativado, 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 MOVED redirecionamento é recebido do servidor, como na situação de um failover, quando todos os slots atendidos pelo antigo nó primário são assumidos pela réplica, ou de fragmentação novamente, quando os slots são movidos do nó primário de origem para o nó primário de destino

A descoberta do cliente geralmente é feita por meio da emissão de um CLUSTER NODE comando CLUSTER SLOT or para o servidor Valkey ou RedisOSS. Recomendamos o CLUSTER SLOT método porque ele retorna o conjunto de intervalos de slots e os nós primários e de réplica associados de volta ao cliente. Isso não exige análise adicional do cliente e é mais eficiente.

Dependendo da topologia do cluster, o tamanho da resposta para o CLUSTER SLOT comando 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 dispara várias solicitações de reconexão e descoberta sem adicionar um recuo exponencial ao tentar novamente. Isso pode fazer com que o OSS servidor Valkey ou Redis não responda por um período prolongado de tempo, com a CPU utilização em 100%. A interrupção é prolongada se cada CLUSTER SLOT comando precisar processar um grande número de nós no barramento do cluster. No passado, observamos várias interrupções de clientes devido a esse comportamento em várias linguagens diferentes, incluindo 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 [Encontrando pontos de extremidade de conexão em 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.

A seguir estão alguns exemplos de código para a lógica de repetição de retrocesso exponencial em redis-py e Lettuce. PHPRedis

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 ativado por meio do `retry_on_timeout` argumento fornecido ao criar um OSS objeto [Redis](#). 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 de lógica de recuo: PHPRedis

PHPRedis tem um mecanismo de repetição integrado que tenta novamente no máximo 10 vezes (não configurável). 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 pull request para implementar nativamente o backoff 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

### Configurando o 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. Alguns OSS comandos do Valkey ou do Redis podem ser computacionalmente mais caros do que outros. Por exemplo, scripts Lua ou EXEC transaçõesMULTI/que contêm vários comandos que devem ser executados atômicamente. 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
- Executando EXEC transaçõesMULTI/ou scripts Lua que consistem em vários comandos individuais do Valkey ou do Redis OSS
- Leitura de valores grandes
- Executando operações de bloqueio, como BLPOP

No caso de uma operação de bloqueioBLPOP, como, a melhor prática é definir o tempo limite do comando para um número menor que o tempo limite do soquete.

A seguir estão exemplos de código para implementar um tempo limite do lado do cliente em redis-py e Lettuce. PHPRedis

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 de tempo limite: PHPRedis

Veja a seguir um exemplo de código comPHPRedis:

```
// 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"); // BLPPOP 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 Valkey e 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 oferecem suporte a mais de 200 comandos, incluindo aqueles para executar scripts Lua. No entanto, quando se trata de scripts Lua, existem várias 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 OSS servidor Valkey ou Redis antes de ser executado. Os scripts Lua não parametrizados são exclusivos, o que pode fazer com que o OSS servidor Valkey ou Redis armazene um grande número de scripts Lua e consuma mais memória. Para mitigar isso, certifique-se de que todos os scripts Lua sejam parametrizados e executados regularmente SCRIPT FLUSH para limpar scripts Lua em cache, se necessário.

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 atômicamente, portanto, podem levar mais tempo para serem concluídos do que um comando normal do Valkey ou do RedisOSS. 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 está sofrendo mutação pode fazer com que o OSS servidor Valkey ou Redis pare de responder por muito 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

Há algumas maneiras pelas quais um script Lua pode continuar gravando novos dados no Valkey ou no Redis, OSS mesmo quando o Valkey ou o OSS Redis terminarem: `maxmemory`

- O script começa quando o OSS servidor Valkey ou Redis está abaixo `maxmemory` e contém várias operações de gravação dentro
- O primeiro comando de gravação do script não está consumindo memória (como `DEL`), seguido por mais operações de gravação que consomem memória
- Você pode mitigar esse problema configurando uma política de despejo adequada no servidor Valkey ou Redis, exceto. OSS `noeviction` Isso permite que o Redis OSS expulse itens e libere memória entre os scripts Lua.

### Armazenamento de grandes itens compostos (Valkey e Redis) OSS

Em alguns cenários, um aplicativo pode armazenar grandes itens compostos no Valkey ou no Redis OSS (como um conjunto de dados de hash de vários GB). Essa não é uma prática recomendada porque geralmente leva a problemas de desempenho no Valkey ou no RedisOSS. Por exemplo, o cliente pode executar um `HGETALL` comando para recuperar toda a coleção de hash de vários GB. Isso pode gerar uma pressão significativa de memória para o OSS servidor Valkey ou Redis, armazenando em buffer o item grande no buffer de saída do cliente. Além disso, para migração de slots no modo de cluster, 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 você precisar acessar vários campos na mesma coleção

atomicamente, poderá usar o MGET comando 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 atomicamente toda a coleção de vários GB no mesmo comando. Um exemplo é usar HMGET comandos HGET or em vez de HGETALL em coleções de hash.

## Configuração do cliente Lettuce (Valkey e Redis) OSS

Esta seção descreve as opções de configuração recomendadas do Java e do Lettuce e como elas se aplicam aos ElastiCache clusters.

As recomendações nesta seção foram testadas com o Lettuce versão 6.2.2.

### Tópicos

- [Exemplo: configuração do Lettuce para o modo de cluster, ativada TLS](#)
- [Exemplo: configuração do Lettuce para o modo de cluster desativada, ativada TLS](#)

### DNSCache Java TTL

A máquina virtual Java (JVM) armazena em cache as pesquisas de DNS nomes. Quando o JVM resolve um nome de host para um endereço IP, ele armazena o endereço IP em cache por um período de tempo especificado, conhecido como (). `time-to-liveTTL`

A escolha do TTL valor é uma troca entre latência e capacidade de resposta às mudanças. Com a versão mais curtaTTLs, DNS os resolvedores notam atualizações no cluster DNS mais rapidamente. 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 TTL for muito baixo, ele aumentará o volume de consultas, o que pode aumentar a latência do seu aplicativo. Embora não haja um TTL valor correto, vale a pena considerar o tempo que você pode esperar até que uma alteração entre em vigor ao definir seu TTL valor.

Como ElastiCache os nós usam entradas de DNS nome que podem mudar, recomendamos que você configure seu JVM com um mínimo TTL de 5 a 10 segundos. Isso garante que, quando o endereço IP de um nó for alterado, seu aplicativo possa receber e usar o novo endereço IP do recurso solicitando a DNS entrada.

Em algumas configurações Java, o JVM padrão TTL é definido para que ele nunca atualize DNS as entradas até que JVM seja reiniciado.

Para obter detalhes sobre como definir seu JVMTTL, consulte [Como definir JVM TTL](#) o.

## 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 DNS busca por isso 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

Habilitar [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 de cliente habilitadas para o modo de cluster

Ative [AutoReconnect](#) quando a conexão for perdida.

Conjunto [CommandTimeout](#). Para obter mais detalhes, consulte a seção “Tempos limite” mais adiante neste tópico.

[nodeFilter](#) Defina para filtrar os nós com falha da topologia. O Lettuce salva todos os nós encontrados na saída de “nós de cluster” (incluindo nós com FAIL status PFAIL/) 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 ele `clusterTopology` é atualizado, o mapa dos nós do barramento do cluster tem um curto período de tempo em que o nó inativo é listado como um FAIL nó, antes de ser completamente



removido da topologia. Durante esse período, o cliente do Lettuce o considera um nóduo saudável e se conecta continuamente 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 melhor usada com `DynamicRefreshSources` definida como verdadeira. 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 verdade) reduz a probabilidade desse problema, pois pelo menos alguns dos nós iniciais devem ter uma visão de topologia atualizada após um failover com o primário recém-promovido.

`ClusterTopologyRefreshOptions`: Opções para controlar a atualização da topologia do cluster do cliente habilitado para o modo de 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 desativado com ElastiCache não é compatível com o Lettuce's. `MasterSlaveTopologyRefresh` 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 [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#).

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.

Habilitar [enablePeriodicRefresh](#). Isso permite atualizações periódicas da topologia do cluster para que o cliente atualize a topologia do cluster 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.

Habilitar [enableAllAdaptiveRefreshTriggers](#). Isso permite a atualização adaptativa da topologia que usa todos os [acionadores](#): `MOVED _REDIRECT`, `_`, `_`, `ASK _REDIRECT`, `PERSISTENT _RECONNECTS`, `UNCOVERED SLOT UNKNOWN NODE`. Os gatilhos de atualização adaptáveis iniciam atualizações de visualização de topologia com base em eventos que ocorrem durante as operações de cluster 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).

Habilitar [closeStaleConnections](#). Isso possibilita fechar conexões obsoletas ao atualizar a topologia do cluster. Só entra em vigor se [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled\(\)](#) é 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.

Habilitar [dynamicRefreshResources](#). Recomendamos ativá-la `dynamicRefreshResources` para clusters pequenos e desativá-la para clusters grandes. `dynamicRefreshResources` permite descobrir

nós do cluster a partir do nó inicial fornecido (por exemplo, endpoint de configuração do 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 verdade, o Lettuce consulta todos os nós para obter a visualização do cluster e, em seguida, 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 DNS resolvidor é baseado no `com.sun.jndi.dns` do Java. `DnsContextFactory`.

Configure [reconnectDelay](#) com recuo exponencial e instabilidade total. O Lettuce tem mecanismos de repetição integrados com base nas estratégias de recuo exponencial. Para obter detalhes, consulte [Exponential Backoff and Jitter](#) no blog de arquitetura. AWS Para obter mais informações sobre a importância de ter uma estratégia de recuo de repetição, consulte as seções de lógica de recuo da [postagem do blog de melhores práticas no blog](#) do AWS banco de dados.

```
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, use um tempo limite maior para comandos que iteram em várias chaves, como `scriptsFLUSHDB`, `FLUSHALL`, `KEYS MEMBERS`, ou `Lua`. Use tempos limite mais curtos para comandos de tecla única `SET`, `GET`, e `HSET`.

### Note

Os tempos limite configurados no exemplo a seguir são para testes que executaram `GET` comandos `SET`/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 o modo de cluster, ativada TLS

#### Note

Os tempos limite no exemplo a seguir são para testes que executaram GET comandos SET/ 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 o modo de cluster desativada, ativada TLS

#### Note

Os tempos limite no exemplo a seguir são para testes que executaram GET comandos SET/ 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);

```

## Configurando um protocolo preferencial para clusters de pilha dupla (Valkey e Redis OSS)

Para OSS clusters Valkey ou Redis habilitados para o modo de cluster, você pode controlar o protocolo que os clientes usarão para se conectar aos nós no cluster com o parâmetro IP Discovery. O parâmetro IP Discovery pode ser definido como IPv4 ou IPv6.

Para OSS clusters Valkey ou Redis, o parâmetro de descoberta de IP define o protocolo IP usado na saída de [slots \(\)](#), [fragmentos de cluster \(\)](#) e nós de [cluster \(\)](#). Esses comandos são usados pelos clientes para descobrir a topologia do cluster. Os clientes usam os comandos `IPs in these` 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 OSS cluster Valkey ou Redis, monitore a saída de `cluster slots`. Depois que todos os nós retornados pelo comando `cluster slots` se reportam 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;
})));
```

## Melhores práticas para clientes (Memcached)

### Configurando seu ElastiCache cliente para balanceamento de carga eficiente (Memcached)

#### Note

Esta seção se aplica a clusters do Memcached multinó autoprotetido.

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 clientes 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](#).

### Uso consistente de hashing PHP com Memcached

O PHP cliente ElastiCache Memcached é um invólucro da biblioteca integrada do Memcached. PHP Por padrão, o hashing consistente é desativado pela biblioteca MemcachedPHP.

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 em http://php.net/manual/en/memcached.configuration.php](http://php.net/manual/en/memcached.configuration.php)PHP. Observe especificamente o parâmetro `memcached.sess_consistent_hash`.

### Uso consistente de hashing. NETcom Memcached

O ElastiCache Memcached. NETclient é um invólucro em torno 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/-Configuration# MemcachedClient](https://github.com/enyim/EnyimMemcached/wiki/-Configuration#MemcachedClient). user-content-memcachedlocator

## Clientes validados com Memcached

Os clientes a seguir foram validados especificamente para funcionar com todas as configurações de tipo de rede compatíveis com o Memcached.

Clientes validados:

- [AWS ElastiCache Cliente de cluster Memcached para PHP — Versão \\*3.6.2](#)
- [AWS ElastiCache Cluster Client Memcached para Java](#) — Master mais recente no Github

## Configurando 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 IP Discovery 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 para clusters ElastiCache (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 de `getServerList`. Depois que a saída dessas funções IPs for reportada para todos os nós no 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.toList());

    Thread.sleep(1000);
}
```

```

} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByName(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.

## TLSclusters de pilha ElastiCache dupla habilitados

Quando TLS está habilitado para ElastiCache clusters, as funções de descoberta de clusters (`cluster slotscluster shards`, e `cluster nodes` para Redis) ou `config get cluster`

para 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 TLS handshake. Isso significa que os clientes não serão afetados pelo parâmetro de descoberta de IP. Para clusters TLS habilitados, o parâmetro IP Discovery não tem efeito no protocolo IP preferido. Em vez disso, o protocolo IP usado será determinado pelo protocolo IP que o cliente prefere ao resolver DNS nomes de host.

## Clientes Java

Ao se conectar a partir de um ambiente Java que suporte ambos IPv4 e IPv6, por padrão, o Java preferirá IPv4 a IPv6 compatibilidade com versões anteriores. No entanto, a preferência do protocolo IP é configurável por meio dos JVM argumentos. Para preferir IPv4, o JVM aceita `-Djava.net.preferIPv4Stack=true` e o preferir IPv6 definem `-Djava.net.preferIPv6Stack=true`. A configuração `-Djava.net.preferIPv4Stack=true` significa que JVM eles não farão mais nenhuma IPv6 conexão. Para Valkey ou RedisOSS, isso inclui aqueles para outros aplicativos que não sejam Valkey e não Redis. OSS

## Preferências de nível de host

Em geral, se o cliente ou o tempo de execução do cliente não fornecerem opções de configuração para definir uma preferência de protocolo IP, ao executar a DNS resolução, o protocolo IP dependerá da configuração do host. Por padrão, a maioria dos hosts prefere IPv6 a IPv4, mas essa preferência pode ser configurada no nível do host. Isso afetará todas as DNS solicitações desse host, não apenas as dos ElastiCache clusters.

## 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 de sua preferência IPv4 ao se conectar a um ElastiCache cluster, atualize a precedência do CIDR intervalo que abrange o cluster IPs para estar acima da precedência das conexões padrão. IPv6 Por padrão, IPv6 as conexões têm precedência de 40. Por exemplo, supondo que o cluster esteja localizado em uma sub-rede com CIDR `172.31.0.0/16`, a configuração abaixo faria com que os clientes preferissem conexões com esse cluster. IPv4

```
label ::1/128      0
label ::/0         1
```



```
label 2002::/16      2
label ::/96          3
label ::ffff:0:0/96 4
label fec0::/10     5
label fc00::/7      6
label 2001:0::/32   7
label ::ffff:172.31.0.0/112 8
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# site-local IPv4 and IPv6 addresses a lookup for a global address would
# see the IPv6 be preferred. The result is a long delay because the
# site-local IPv6 addresses cannot be used while the IPv4 address is
# (at least for the foreseeable future) NATed. We also treat Teredo
# tunnels special.
#
# precedence <mask> <value>
# Add another rule to the RFC 3484 precedence table. See section 2.1
# and 10.3 in RFC 3484. The default is:
#
precedence ::1/128      50
precedence ::/0        40
precedence 2002::/16   30
precedence ::/96       20
precedence ::ffff:0:0/96 10
precedence ::ffff:172.31.0.0/112 100
```

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 IPv4 conexões em vez de IPv6 conexões para o CIDR intervalo especificado. Por exemplo, supor que o cluster esteja em uma sub-rede com a CIDR execução `172.31.0. 0:0 /16` `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` resultaria na tabela de precedência a seguir, o que faria com que os clientes preferissem se conectar ao cluster. IPv4

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...
```

```
Precedence Label Prefix
-----
```

```
100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```

## Gerenciando a memória reservada para Valkey e Redis OSS

A memória reservada é a memória reservada para o uso não relacionado a dados. Ao realizar um backup ou um failover, o Valkey e o Redis OSS usam a memória disponível para gravar operações de gravação no cluster enquanto os dados do cluster são gravados no arquivo.rdb. Se você não tiver memória suficiente disponível para todas as gravações, o processo falhará. A seguir, você encontrará informações sobre as opções para gerenciar a memória reservada para ElastiCache (RedisOSS) 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 failovers do que se estivesse executando o Redis 2.8.22 ou posterior. OSS Esse requisito se deve às diferentes maneiras pelas quais ElastiCache (RedisOSS) implementa o processo de backup. A regra geral é reservar metade do `maxmemory` valor de um tipo de nó para a OSS sobrecarga do Redis para versões anteriores à 2.8.22 e um quarto para as versões 2.8.22 e posteriores do Redis. OSS

Devido às diferentes maneiras de ElastiCache implementar o processo de backup e replicação, a regra geral é reservar 25% do `maxmemory` valor de um tipo de nó usando o parâmetro. `reserved-memory-percent` Esse é o valor padrão e é recomendado para a maioria dos casos.

Quando os tipos de instância micro e pequena com capacidade de intermitência estão operando perto dos `maxmemory` limites, eles podem experimentar o uso de swap. Para melhorar a confiabilidade operacional desses tipos de instância durante o backup, a replicação e o tráfego intenso, recomendamos aumentar o valor do `reserved-memory-percent` parâmetro em até 30% nos tipos de instâncias pequenas e até 50% nos tipos de microinstância.

Para cargas de trabalho com muita gravação em ElastiCache clusters com armazenamento de dados em camadas, recomendamos aumentar `reserved-memory-percent` até 50% da memória disponível do nó.

Para obter mais informações, consulte as informações a seguir.

- [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#)
- [Como a sincronização e o backup são implementados](#)
- [Hierarquização de dados em ElastiCache](#)

## Parâmetros para gerenciar memória reservada

A partir de 16 de março de 2017, a Amazon ElastiCache fornece dois parâmetros mutuamente exclusivos para gerenciar sua OSS memória Valkey ou Redis, e. `reserved-memory` `reserved-memory-percent` Nenhum desses parâmetros faz parte da distribuição Valkey ou RedisOSS.

Dependendo de quando você se tornou ElastiCache cliente, um ou outro desses parâmetros é o parâmetro padrão de gerenciamento de memória. Esse parâmetro se aplica quando você cria um novo OSS cluster ou grupo de replicação Valkey ou Redis e usa um grupo de parâmetros padrão.

- Para clientes que começaram antes de 16 de março de 2017 — Quando você cria um OSS cluster ou grupo de replicação do Redis usando o grupo de parâmetros padrão, seu parâmetro de gerenciamento de memória é. `reserved-memory` Nesse caso, zero (0) bytes de memória são reservados.
- Para clientes que começaram em ou após 16 de março de 2017 — Quando você cria um OSS cluster ou grupo de replicação Valkey ou Redis usando o grupo de parâmetros padrão, seu parâmetro de gerenciamento de memória é. `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 OSS memória Valkey ou Redis, talvez você prefira usar aquele que não é seu 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. Em seguida, você pode usar o grupo de parâmetros personalizado sempre que criar um novo OSS cluster ou grupo de replicação Valkey ou Redis. Para clusters ou grupos de replicação existentes, você pode modificá-los para usar o `parameter group` personalizado.

Para obter mais informações, consulte as informações a seguir.

- [Especificação do parâmetro de gerenciamento de memória reservada](#)

- [Criação de um grupo de ElastiCache parâmetros](#)
- [Modificando um grupo de ElastiCache parâmetros](#)
- [Modificando um cluster ElastiCache](#)
- [Modificação de um grupo de replicação](#)

## O parâmetro reserved-memory

Antes de 16 de março de 2017, todo o gerenciamento de memória reservada ElastiCache (RedisOSS) era feito usando o parâmetro `reserved-memory`. O valor padrão de `reserved-memory` é 0. Esse padrão não reserva memória para a OSS sobrecarga do Valkey ou do Redis e permite que o Valkey ou o Redis consumam toda a OSS 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. Nesse grupo de parâmetros personalizados, você define `reserved-memory` um valor apropriado para a OSS versão Valkey ou Redis em execução no seu cluster e no tipo de nó do cluster. Para ter mais informações, consulte [De quanta memória reservada você precisa?](#)

O parâmetro `reserved-memory` é específico ElastiCache e não faz parte da OSS distribuição geral do Redis.

O procedimento a seguir mostra como usar `reserved-memory` para gerenciar a memória em seu cluster Valkey ou RedisOSS.

## 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 ElastiCache parâmetros](#).

```
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 devem ser reservados para a sobrecarga do Valkey ou do RedisOSS. Você pode encontrar o valor de `maxmemory` para o seu tipo de nó em [Parâmetros específicos do tipo de OSS nó Redis](#).

3. Modifique o parameter group personalizado para que o parâmetro `reserved-memory` seja o número de bytes que você calculou na etapa anterior. O AWS CLI exemplo a seguir pressupõe que você esteja executando uma versão do Redis OSS anterior à 2.8.22 e precise reservar metade do nó. `maxmemory` Para obter mais informações, consulte [Modificando um grupo de ElastiCache parâmetros](#).

```
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 OSS cluster ou grupo de replicação do Redis para usar seu grupo de parâmetros personalizado.

O CLI exemplo a seguir modifica o cluster `my-redis-cluster` para usar o grupo de parâmetros personalizado `redis28-m3x1` começando imediatamente. Para obter mais informações, consulte [Modificando um cluster ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

O CLI exemplo a seguir modifica o grupo de replicação `my-redis-repl-grp` para usar o grupo de parâmetros personalizado `redis28-m3x1` começando imediatamente. 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 reserved-memory-percent parâmetro

Em 16 de março de 2017, a Amazon ElastiCache apresentou o parâmetro `reserved-memory-percent` e o disponibilizou em todas as versões do ElastiCache (RedisOSS). 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 ElastiCache e não faz parte da OSS distribuição geral do Redis.

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 [Hierarquização de dados em ElastiCache](#).

Para reservar memória usando `reserved-memory-percent`

Para usar `reserved-memory-percent` para gerenciar a memória em seu cluster ElastiCache (RedisOSS), faça o seguinte:

- Se você estiver executando o Redis OSS 2.8.22 ou posterior, atribua o grupo de parâmetros padrão ao seu 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 `reserved-memory-percent` do que os 25% padrão. 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 ElastiCache parâmetros](#).

```
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 [Modificando um grupo de ElastiCache parâmetros](#).

```
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 OSS cluster ou grupo de replicação do Redis executando uma versão do Redis OSS anterior à 2.8.22.

O CLI exemplo a seguir modifica o OSS cluster Redis `my-redis-cluster` para usar o grupo de parâmetros personalizado `redis28-50` começando imediatamente. Para obter mais informações, consulte [Modificando um cluster ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

O CLI exemplo a seguir modifica o grupo de OSS replicação do Redis `my-redis-repl-grp` para usar o grupo de parâmetros personalizado começando imediatamente. `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 era um ElastiCache cliente atual em 16 de março de 2017, seu parâmetro padrão de gerenciamento de memória reservada é `reserved-memory` com zero (0) bytes de memória reservada. Se você se tornou ElastiCache cliente depois de 16 de março de 2017, seu parâmetro padrão de gerenciamento de memória reservada é `reserved-memory-percent` com 25% da memória do nó reservada. Isso é verdade independentemente de quando você criou seu cluster ElastiCache (RedisOSS) ou grupo de replicação. No entanto, você pode alterar seu parâmetro de gerenciamento de memória reservada usando o AWS CLI ou ElastiCache API.

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 ElastiCache parâmetros](#).

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 CLI exemplo a seguir modifica o grupo de parâmetros 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 [Modificando um grupo de ElastiCache parâmetros](#).

```
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 CLI exemplo a seguir modifica o grupo de parâmetros personalizado `redis32-m3x1` para que ele o 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 `maxmemory` de um cache `m3.xlarge`. Para obter mais informações, consulte [Modificando um grupo de ElastiCache parâmetros](#).

```
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 autoprojeto do Valkey e do Redis OSS

O uso do Multi-AZ, a memória suficiente, o redimensionamento do cluster e a minimização do tempo de inatividade são conceitos úteis que você deve ter em mente ao trabalhar com clusters autoprojeto no Valkey ou no Redis. OSS É recomendável analisar e seguir essas práticas recomendadas.

### Tópicos

- [Minimização do tempo de inatividade com multi-AZ](#)
- [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#)
- [Redimensionamento de cluster on-line](#)
- [Minimização do tempo de inatividade durante a manutenção](#)

### Minimização do tempo de inatividade com multi-AZ

Há vários casos em que o ElastiCache Valkey ou o Redis OSS podem precisar substituir um nó primário; isso inclui certos tipos de manutenção planejada e o evento improvável de uma falha no nó primário ou na 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, pois ElastiCache trataremos disso 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.

Consulte [Minimizando o tempo de inatividade ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#) para saber mais sobre o Multi-AZ e como minimizar o tempo de inatividade.

## Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS

Instantâneos e sincronizações no Valkey 7.2 e posterior e no Redis versão 2.8.22 e posterior OSS

O Valkey tem suporte padrão para instantâneos e sincronizações. O Redis OSS 2.8.22 introduz um processo de salvamento sem bifurcação que permite alocar mais memória para o uso do aplicativo sem incorrer no aumento do uso de swap durante sincronizações e salvamentos. Para obter mais informações, consulte [Como a sincronização e o backup são implementados](#).

OSS Instantâneos e sincronizações do Redis antes da versão 2.8.22

Quando você trabalha com ElastiCache (RedisOSS), 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 ativar o recurso de anexar somente arquivo (AOF) para 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 do processo. 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 OSS cluster Redis.

Processo de gravação em segundo plano e uso de memória com Valkey e Redis OSS

Sempre que um processo de gravação em segundo plano é chamado, o Valkey e o Redis OSS bifurcam seu processo (lembre-se de que esses mecanismos são de thread único). Uma bifurcação persiste seus dados no disco em um arquivo de instantâneo Redis OSS .rdb. A outra bifurcação atende a todas as operações de leitura e gravação. Para garantir que seu instantâneo seja um

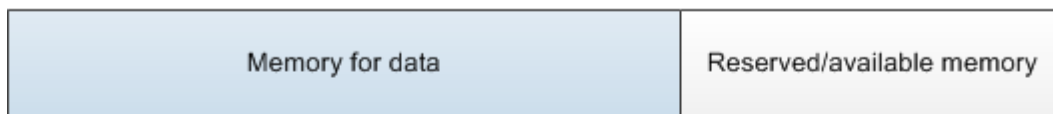
point-in-time instantâneo, todas as atualizações e adições de dados são gravadas em uma área da memória disponível separada da área de dados.

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:

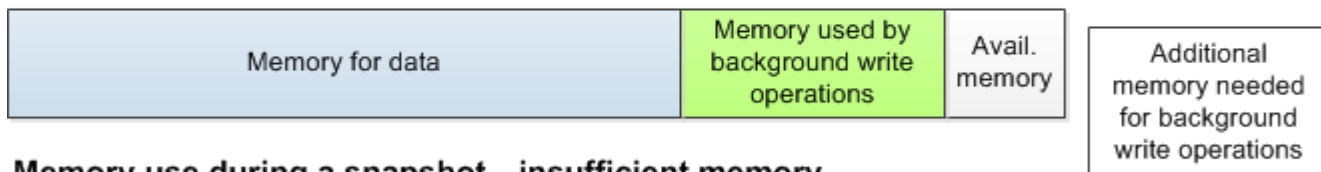
- Seu aplicativo realiza muitas operações de gravação, exigindo assim uma grande quantidade de memória disponível para aceitar os dados novos ou atualizados.
- Você tem pouca memória disponível para gravar dados novos ou atualizados.
- Você tem um grande conjunto de dados que demora muito para persistir 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 autoprotetidos](#).

Para obter mais informações sobre como o Valkey e o Redis OSS realizam instantâneos, consulte <http://valkey.io>.

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Escolhendo regiões e zonas de disponibilidade para 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 a consumida pelas operações de gravação durante o processo. O pior cenário é que, durante a operação de gravação em segundo plano, cada registro seja atualizado e alguns novos registros sejam adicionados ao cache. Por isso, recomendamos que você `reserved-memory-percent` defina como 50 (50 por cento) para OSS as versões do Redis anteriores à 2.8.22 ou 25 (25 por cento) para o Valkey e todas as versões 2.8.22 e posteriores do RedisOSS.

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 consuma toda OSS a memória máxima 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 do tipo de OSS nó Redis](#).

Você também pode usar o `reserved-memory` parâmetro para reduzir a quantidade de memória usada na caixa.

Para obter mais informações sobre os parâmetros específicos do Valkey e do Redis em, consulte. ElastiCache [Parâmetros Valkey e Redis OSS](#)

Para obter informações sobre como criar e modificar parameter groups, consulte [Criação de um grupo de ElastiCache parâmetros](#) e [Modificando um grupo de ElastiCache parâmetros](#).

## 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 isso, recomendamos manter a CPU utilização abaixo de 80% em instâncias de vários núcleos e menos de 50% em instâncias de núcleo único durante a refragmentação. Monitore as métricas ElastiCache (RedisOSS) e inicie a refragmentação antes que seu aplicativo 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 escalar seu cluster, ElastiCache replicará automaticamente as funções carregadas em um dos nós existentes (selecionados aleatoriamente) para o (s) novo (s) nó (s). Se seu cluster tiver Valkey 7.2 e superior, ou Redis OSS 7.0 ou superior, e seu aplicativo usar [Functions](#), recomendamos carregar todas as suas funções em todos os fragmentos antes de escalar 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.

- Slots com itens grandes não são migrados. Especificamente, slots com itens maiores do que 256 MB após a serialização não são migrados.
- Os comandos FLUSHALL e FLUSHDB não são compatíveis em scripts Lua durante uma operação de reestilhecimento. Antes do Redis OSS 6, o BRPOPLPUSH comando não era suportado se ele operasse 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 AutoFailover estiver habilitado no cluster, o nó primário poderá mudar. 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 a função AutoFailover desativada, a função do nó não muda. No entanto, o tempo de inatividade em operações gerenciadas ou não gerenciadas é maior em comparação com AutoFailover clusters habilitados.

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 encontrará estratégias para preencher e manter o 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

- [Carregamento lento](#)
- [Gravação simultânea](#)
- [Adicionando TTL](#)
- [Tópicos relacionados da](#)

## 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.

ElastiCache A Amazon é um armazenamento de valores-chave na memória que fica entre seu aplicativo e o armazenamento de dados (banco de dados) que ele acessa. Sempre que seu aplicativo solicita dados, ele primeiro faz a solicitação ao ElastiCache cache. Se os dados existirem no cache e forem atuais, ElastiCache retornará os dados para seu aplicativo. 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 [Adicionando 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 recuperarem 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 time to live \(TTL\)](#), você pode minimizar o desperdício de espaço.

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"})
```

## Adicionando 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 time to live (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.

Time to live (TTL) é um valor inteiro que especifica o número de segundos até a chave expirar. O Valkey ou o Redis OSS podem especificar segundos ou milissegundos para esse 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 e Redis ou OSS os comandos set Memcached](#).

### TTL exemplos de pseudocódigo

A seguir está um exemplo de pseudocódigo da lógica de gravação 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
```

A seguir está um exemplo de pseudocódigo da 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 da

- [Armazenamento de dados na memória](#)
- [Escolha de um mecanismo e versão](#)
- [Dimensionamento ElastiCache](#)

## Gerenciando seu cluster autoprojeto em ElastiCache

ElastiCache oferece duas opções de implantação, armazenamento em cache sem servidor e clusters autoprojeto. Cada um tem suas próprias capacidades e requisitos.

Esta seção contém tópicos para ajudá-lo a gerenciar seus clusters projetados por você mesmo.

 Note

Esses tópicos não se aplicam ao ElastiCache Serverless.

## Tópicos

- [Clusters Auto Scaling 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](#)
- [Gerenciando a manutenção ElastiCache do cluster](#)
- [Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros](#)

## Clusters Auto Scaling Valkey e Redis OSS

### Pré-requisitos

ElastiCache O Auto Scaling está limitado ao seguinte:

- Clusters Valkey ou Redis OSS (modo de cluster ativado) executando Valkey 7.2 em diante ou executando o mecanismo Redis versão 6.0 em diante OSS
- Clusters de armazenamento de dados em camadas (modo de cluster ativado) executando Valkey 7.2 em diante ou executando o mecanismo Redis OSS versão 7.0.7 em diante
- Tamanhos de instância - GrandeXLarge, 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.

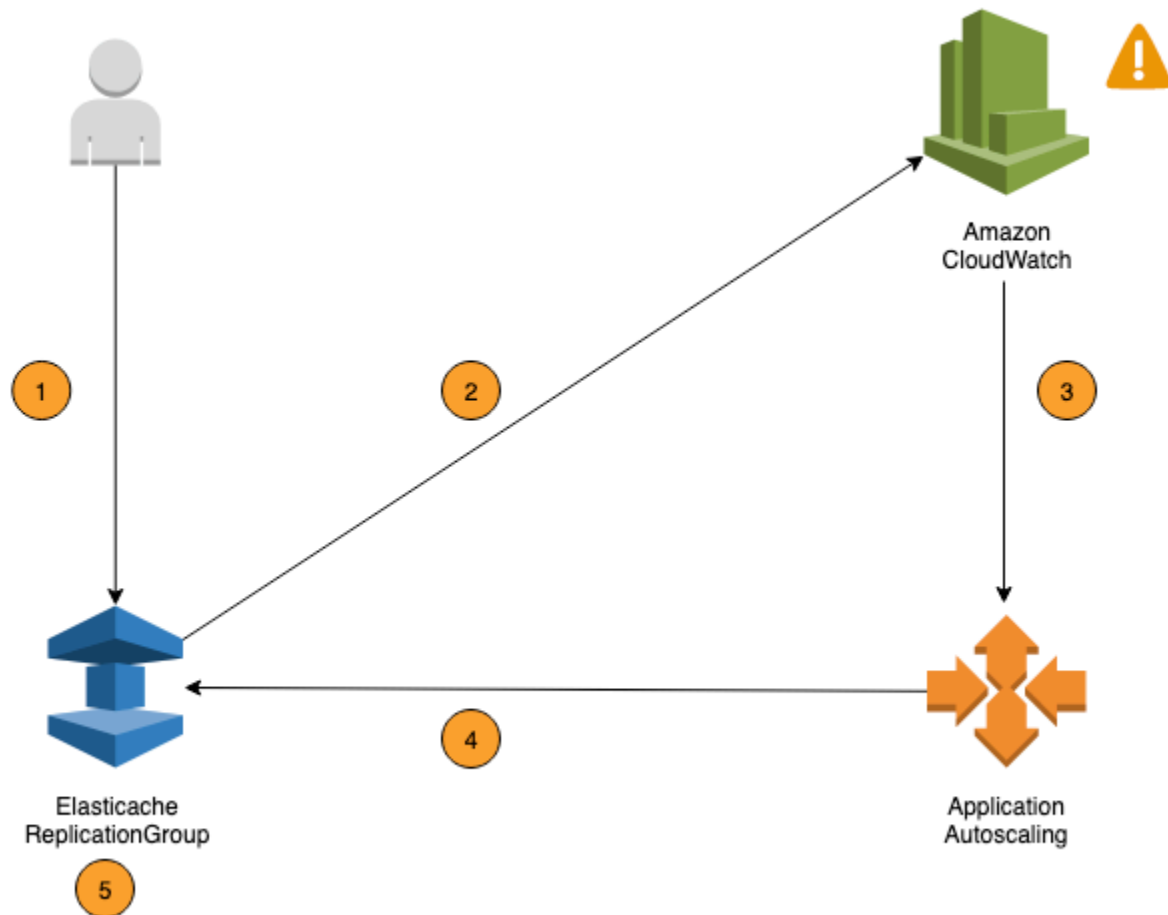
Você pode usar o AWS Management Console para aplicar uma política de escalabilidade com base em uma métrica predefinida. Uma `predefined metric` é definida em uma enumeração, para que você possa especificá-la por nome no código ou usá-la no AWS Management Console. As métricas personalizadas não estão disponíveis para seleção ao usar o AWS Management Console. Como alternativa, você pode usar o Application Auto Scaling AWS CLI ou o Application Auto Scaling API para aplicar uma política de escalabilidade com base em uma métrica predefinida ou personalizada.

ElastiCache com Valkey ou Redis 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.
- **Replicas (Réplicas):** adicionar/remover automaticamente réplicas no cluster semelhantes às operações manuais de aumentar/diminuir réplicas. ElastiCache com o escalonamento OSS automático Valkey ou Redis, adiciona/remove réplicas uniformemente em todos os fragmentos do cluster.

ElastiCache com Valkey ou Redis OSS suporta os seguintes tipos de políticas de escalabilidade automática:

- [Políticas de escalabilidade de rastreamento de destino](#): aumenta ou diminui o número de tarefas que o serviço executa com base em um valor de destino 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 aplicativos ElastiCache com escalabilidade OSS automática Valkey ou Redis](#) — Aumente ou diminua o número de fragmentos/réplicas que seu serviço executa com base na data e hora.



As etapas a seguir resumem o processo de escalonamento OSS automático ElastiCache com Valkey ou Redis, conforme mostrado no diagrama anterior:

1. Você cria uma política de escalabilidade ElastiCache automática para seu grupo de replicação.
2. ElastiCache o escalonamento automático com Valkey ou Redis OSS 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 com o escalonamento OSS automático Valkey ou Redis, emite uma solicitação de modificação para ajustar a capacidade do cluster.



5. ElastiCache com Valkey ou Redis OSS processa a solicitação de modificação, aumentando (ou diminuindo) dinamicamente a capacidade de fragmentos/réplicas do cluster para que ela se aproxime da utilização desejada.

Para entender como o ElastiCache Valkey ou o Redis Auto OSS 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 CPU utilização do `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 com Valkey ou Redis Auto OSS 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 rastreamento de metas de escalonamento automático busca manter a utilização alvo igual ou próxima ao valor escolhido a longo prazo.

## Políticas de Auto Scaling

Uma política de escalabilidade tem os seguintes componentes:

- Uma métrica alvo — A CloudWatch métrica que, ElastiCache com o Valkey ou o Redis OSS Auto Scaling, usa para determinar quando e quanto 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 que a capacidade máxima configurada, fazemos isso `MaxCapacity` durante `scaleIn` a criação da política. Da mesma forma, se a capacidade atual for menor que a capacidade mínima configurada, nós vamos `scaleOut` para `MinCapacity` o.

- 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 ao serviço — uma função do AWS Identity and Access Management (IAM) vinculada a um serviço específico AWS . Uma função vinculada ao serviço inclui todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome. ElastiCache com Valkey ou RedisOSS, o Auto Scaling 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

Nesse tipo de política, uma métrica predefinida ou personalizada e um valor alvo para a métrica são especificados em uma configuração de política de escalabilidade de rastreamento de metas. ElastiCache com Valkey ou Redis, o Auto OSS Scaling cria e CloudWatch gerencia alarmes que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor alvo. 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 CPU utilização em, ou perto de, uma porcentagem específica de utilização, como 70 por cento.

**Note**

Para cada cluster, você pode criar somente uma política de Auto Scaling para cada métrica de destino.

## Capacidades mínima e máxima

### Estilhaços

Você pode especificar o número máximo de fragmentos que podem ser escalados ElastiCache com o escalonamento automático Valkey ou Redis. 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 escalonamento 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 ElastiCache com o OSS escalonamento automático Valkey ou Redis. Esse valor deve ser igual ou menor que 5. Você também pode especificar o número mínimo de réplicas a serem gerenciadas pelo escalonamento 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**

ElastiCache com as políticas de OSS auto scaling do Valkey ou do Redis, aumente a capacidade do cluster até que ele atinja o tamanho máximo definido ou até que os limites de 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 se tornar zero, ElastiCache com Valkey ou Redis, escale OSS automaticamente 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 retarda a exclusão de fragmentos/réplicas em seu OSS cluster ElastiCache com Valkey ou Redis para solicitações de expansão e a criação de fragmentos/réplicas para solicitações de expansão. Você pode especificar os seguintes desaquecimentos:

- Uma atividade de escalabilidade reduz 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.
- Uma atividade de expansão 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 escalabilidade horizontal estão sempre habilitadas para que a política de escalabilidade possa ser criada ElastiCache com fragmentos/réplicas do Valkey ou do RedisOSS, conforme necessário.

## IAMPermissões necessárias para o Auto Scaling

ElastiCache com Valkey ou Redis, o Auto OSS Scaling é possível graças a uma combinação do CloudWatch,, e ElastiCache do Application Auto Scaling. APIs Os clusters são criados e atualizados com ElastiCache (RedisOSS), os alarmes são criados com e as políticas de escalabilidade são criadas com CloudWatch o Application Auto Scaling. Além das IAM permissões padrão para criar e atualizar clusters, o IAM usuário que acessa as configurações do ElastiCache Auto Scaling deve ter as permissões apropriadas para os serviços que oferecem suporte ao escalonamento dinâmico. IAMos usuários devem ter permissões para usar as ações mostradas no exemplo de política a seguir:

```
{
  "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 OSS automático ElastiCache com Valkey ou Redis também precisa de permissão para descrever seus clusters e CloudWatch alarmes, além de permissões para modificar sua capacidade ElastiCache alvo em seu nome. Se você habilitar o Auto Scaling para 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 para escalonamento automático ElastiCache \(Redis\) no Guia do usuário do Application OSS Auto Scaling](#).

## Práticas recomendadas de escalabilidade automática

Antes de se registrar no Auto Scaling, recomendamos o seguinte:

1. Use apenas uma métrica de rastreamento — identifique se seu cluster tem CPU cargas de trabalho com uso intenso de dados e use uma métrica predefinida correspondente para definir a política de escalabilidade.
  - MotorCPU: `ElastiCachePrimaryEngineCPUUtilization` (dimensão do fragmento) 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 com Valkey ou Redis, o OSS Auto Scaling expandirá a meta escalável se alguma política de rastreamento de alvos estiver pronta para ser expandida, mas só será ampliada 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 workload é determinística (alcança um nível alto/baixo em um momento específico), recomendamos usar a escalabilidade programada e configurar sua capacidade de destino 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. Desabilite a redução da escala horizontalmente: a autoescalabilidade no monitoramento do objetivo é mais adequada para clusters com aumento/diminuição gradual da workload, já que picos/queda nas métricas podem desencadear oscilações consecutivas de aumento/redução da escala horizontalmente. 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 sua aplicação: recomendamos que você teste sua aplicação com suas workloads mínimas e máximas estimadas para determinar os mínimos e máximos absolutos de fragmentos/réplicas necessários para o cluster enquanto cria políticas de escalabilidade a fim de 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 você ainda não tem certeza de qual valor escolher, recomendamos começar com o valor mínimo de métrica predefinida compatível.

7. AutoScaling on Target Tracking é mais adequado para clusters com distribuição uniforme de cargas de trabalho na dimensão de fragmentos/réplicas. Ter distribuição não uniforme pode levar a:

- Escalar quando não for necessário devido a picos/quedas de workload em alguns fragmentos/réplicas quentes.
- Não escalar quando necessário devido à média geral perto do destino, mesmo tendo fragmentos/réplicas quentes.

#### Note

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 seu cluster tiver Valkey ou Redis OSS 7.0 ou superior e seu aplicativo usar [Funções](#), recomendamos carregar todas as suas funções em todos os fragmentos antes de escalar para que seu cluster não tenha funções diferentes em fragmentos diferentes.

Depois de se registrar AutoScaling, observe o seguinte:

- Há limitações nas configurações compatíveis com a autoescalabilidade. Portanto, recomendamos que não altere a configuração de um grupo de replicação registrado para escalabilidade automática. Veja os exemplos a seguir:
  - Modificação manual do tipo de instância para tipos sem suporte.
  - Associação do grupo de replicação a um datastore global.
  - Alteração do parâmetro ReservedMemoryPercent.
  - Aumento/diminuição manual dos fragmentos/réplicas além da capacidade mínima e máxima configurada durante a criação da política.

## Uso do Auto Scaling com fragmentos

Com ElastiCache isso, AutoScaling você pode usar políticas de rastreamento e programadas com seu mecanismo Valkey ou RedisOSS.

A seguir, são fornecidos detalhes sobre o rastreamento de metas e as políticas programadas e como aplicá-las usando o AWS Management Console AWS CLI APIs e.



## Tópicos

- [Políticas de escalabilidade de rastreamento de destino](#)
- [Adicionar uma política de escalabilidade](#)
- [Registro de um destino escalável](#)
- [Definir uma política de ajuste de escala](#)
- [Desabilitar a atividade de redução](#)
- [Aplicar uma política de escalabilidade](#)
- [Editar uma política de escalabilidade](#)
- [Excluir uma política de escalabilidade](#)
- [Use AWS CloudFormation para políticas de 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. ElastiCache com Valkey ou Redis, o Auto OSS Scaling cria e gerencia CloudWatch os 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 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 CPU utilização no valor alvo especificado ou próximo a ele.

### Métricas predefinidas

Uma métrica predefinida é uma estrutura que se refere a um nome, dimensão e estatística (average) específicos de uma determinada CloudWatch métrica. Sua política de ajuste de escala automático define as seguintes métricas predefinidas para seu cluster:

| Nome da métrica predefinida                               | CloudWatch Nome da métrica                     | CloudWatch Dimensão métrica                         | Tipos de instância inelegíveis |
|-----------------------------------------------------------|------------------------------------------------|-----------------------------------------------------|--------------------------------|
| ElastiCachePrimaryEngineCPUUtilization                    | EngineCPUUtilization                           | ReplicationGroupId, Função = Primária               | Nenhum                         |
| ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage | DatabaseCapacityUsageCountedForEvictPercentage | Métricas do grupo de OSS replicação Valkey ou Redis | Nenhum                         |
| ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage   | DatabaseMemoryUsageCountedForEvictPercentage   | Métricas do grupo de OSS replicação Valkey ou Redis | R6gd                           |

Os tipos de instância em camadas de dados não podem ser usados `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, pois esses tipos de instância armazenam dados na memória e SSD. O caso de uso esperado para instâncias em camadas de dados é ter 100% de uso de memória e ser SSD preenchida conforme necessário.

#### Crítérios 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. ElastiCache com Valkey ou Redis OSS expande seus fragmentos de cluster em uma contagem igual ao maior de dois números: variação percentual do Target e 20% dos fragmentos atuais. Para aumentar a escala, ElastiCache

não aumentará automaticamente a menos que o valor geral da métrica esteja abaixo de 75 por cento da meta definida.

Para um exemplo de aumento de escala na horizontal, se você tiver 50 fragmentos e

- se seu Target violar em 30 por cento, ElastiCache com o Valkey ou o Redis, a OSS escala aumenta em 30 por cento, o que resulta em 65 fragmentos por cluster.
- se o Target violar em 10 por cento, ElastiCache com o Valkey ou o Redis OSS escalável por padrão. Mínimo de 20 por cento, o que resulta em 60 fragmentos por cluster.

Para um exemplo de aumento de escala, se você selecionou um valor alvo de 60%, ElastiCache com Valkey ou Redis, OSS não haverá escalabilidade automática até que a métrica seja menor ou igual a 45 por cento (25 por cento abaixo da meta de 60 por cento).

### 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 metas para escalar quando a métrica especificada está abaixo do valor alvo. ElastiCache com Valkey ou Redis OSS expande os fragmentos em um desvio mínimo de 20% do alvo 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, ElastiCache com o Valkey ou o RedisOSS, o Auto Scaling sempre age de forma conservadora, arredondando para cima ou para baixo ao determinar quanta capacidade adicionar ou remover. 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 de rastreamento de destino para um OSS cluster ElastiCache com Valkey ou Redis, desde que cada uma delas use uma métrica diferente. A intenção do Auto Scaling ElastiCache (RedisOSS) é sempre priorizar a disponibilidade, portanto, seu comportamento difere dependendo se as políticas de rastreamento de destino estão prontas

para expansão horizontal ou ampliada. 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 CloudWatch alarmes que, ElastiCache com o Valkey ou o Redis OSS Auto Scaling, gerencia para uma política de escalabilidade de rastreamento de destino. ElastiCache O Auto Scaling exclui os alarmes automaticamente quando você exclui a política de escalabilidade.
- ElastiCache O Auto Scaling não impede que você modifique manualmente os fragmentos do cluster. Esses ajustes manuais não afetam nenhum CloudWatch alarme existente associado à política de escalabilidade, mas podem afetar as métricas que podem acionar esses CloudWatch alarmes.
- Esses CloudWatch alarmes gerenciados pelo Auto Scaling são definidos pela AVG métrica em todos os fragmentos do cluster. Assim, ter fragmentos quentes pode resultar em qualquer cenário de:
  - dimensionamento quando não é necessário devido à carga em alguns fragmentos quentes que acionam um alarme CloudWatch
  - não é escalável quando necessário devido à agregação de AVG todos os fragmentos que afeta o alarme de não violação.
- ElastiCache com Valkey ou Redis, os limites OSS padrão de 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 no seu VPC, o que é necessário durante a expansão. Para obter mais informações, consulte [Interfaces de rede elástica](#).
- Se não houvesse capacidade suficiente disponível EC2, o ElastiCache Auto Scaling não escalaria e seria adiado até que a capacidade estivesse disponível.
- ElastiCache (RedisOSS) O Auto Scaling durante a expansão não removerá fragmentos com slots com um tamanho de item maior que 256 MB após a 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 AWS Management Console

Para adicionar uma política de Auto Scaling a um cluster ElastiCache com Valkey ou Redis OSS

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon 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:
  - CPUUtilização primária para criar uma política com base na CPU utilização média.
  - 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 Capacidade inclui memória e SSD utilização para instâncias em camadas de dados e utilização de 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 escalonamento automático manterá esse valor para a métrica alvo selecionada em seus ElastiCache fragmentos:
  - CPUUtilização primária: mantém o valor alvo da EngineCPUUtilization métrica 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 o AWS CLI para modificar os valores do tempo de recarga.
11. Em Capacidade mínima, digite o número mínimo de fragmentos que a política de ElastiCache Auto Scaling deve manter.
12. Em Capacidade máxima, digite o número máximo de fragmentos que a política de ElastiCache Auto Scaling deve manter. Esse valor deve ser igual ou menor que 250.
13. Escolha Criar.

## Registro de um destino escalável

Antes de usar o Auto Scaling com um cluster ElastiCache com Valkey ou Redis, você registra seu OSS cluster com o Auto Scaling. ElastiCache Você faz isso para definir a dimensão e os limites de escalabilidade a serem aplicados a esse cluster. ElastiCache o auto scaling dimensiona dinamicamente o cluster ao longo da dimensão `elasticache:replication-group:NodeGroups` escalável, que representa o número de fragmentos do cluster.

## Usando o AWS CLI

Para registrar seu no ElastiCache Valkey ou no OSS cluster Redis, use o [register-scalable-target](#) comando com os seguintes parâmetros:

- `--service-namespace`: defina o valor como `elasticache`
- `--resource-id`— O identificador do recurso para o cluster. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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 ElastiCache escalonamento automático. 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 ElastiCache escalonamento automático. 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, você registra um OSS cluster ElastiCache com Valkey ou Redis chamado `myscalecluster`. 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/myscalecluster \  
  --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/myscalecluster ^  
  --scalable-dimension elasticache:replication-group:NodeGroups ^  
  --min-capacity 1 ^  
  --max-capacity 10 ^  

```

## Usando o API

Para registrar seu ElastiCache cluster, use o [register-scalable-target](#) comando com os seguintes parâmetros:

- `ServiceNamespace` — Defina esse valor como `elasticache`.
- `resourceID` — O identificador do recurso para o cluster. ElastiCache Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster, por exemplo `replication-group/myscalecluster`.
- `ScalableDimension` — Defina esse valor como `elasticache:replication-group:NodeGroups`.
- `MinCapacity` — O número mínimo de fragmentos a serem gerenciados pelo ElastiCache escalonamento automático. 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 ElastiCache escalonamento automático. 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, você registra um OSS cluster ElastiCache com Valkey ou Redis chamado `myscalecluster` com o Application Auto Scaling. API 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/myscalecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

## Definir uma política de ajuste de escala

Uma configuração de política de escalabilidade de rastreamento de metas é representada por um JSON bloco no qual as métricas e os valores-alvo são definidos. Você pode salvar uma configuração de política de escalabilidade como um JSON bloco em um arquivo de texto. Você usa esse arquivo de texto ao invocar o AWS CLI ou o Application API Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de políticas, consulte [TargetTrackingScalingPolicyConfiguration](#) na Application Auto API Scaling Reference.

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

Usando métricas predefinidas, você pode definir rapidamente uma política de escalabilidade de rastreamento de metas para um OSS cluster ElastiCache com Valkey ou Redis que funcione com rastreamento de alvos no Auto Scaling (Redis ElastiCache ). OSS

Atualmente, é ElastiCache compatível com as seguintes métricas predefinidas no NodeGroup Auto Scaling:

- `ElastiCachePrimaryEngineCPUUtilization`— O valor médio da `EngineCPUUtilization` métrica em CloudWatch todos os nós primários do cluster.
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— O valor médio da `DatabaseMemoryUsageCountedForEvictPercentage` métrica em CloudWatch todos os nós primários do cluster.
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— O valor médio da `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` métrica em CloudWatch todos os nós primários do 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 um `PredefinedMetricSpecification` para a métrica predefinida e um `TargetValue` para o valor alvo dessa métrica.

## Example

O exemplo a seguir descreve uma configuração de política típica para escalonamento de rastreamento de destinos para um cluster ElastiCache com Valkey ou Redis. OSS Nessa configuração, a métrica `ElastiCachePrimaryEngineCPUUtilization` predefinida é usada para ajustar o cluster com base em uma CPU utilização média de 40% em todos os nós primários do 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 ElastiCache métrica que mude em proporção à escala. Nem todas as ElastiCache métricas funcionam para o rastreamento de metas. 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 ElastiCache (RedisOSS) com base em uma CPU utilização média de 50% em todos os fragmentos em um cluster chamado. `my-db-cluster`

```
{
  "TargetValue": 50,
  "CustomizedMetricSpecification":
  {
    "MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      {
        "Name": "RelicationGroup","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 do Application Auto Scaling. API

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica `ElastiCachePrimaryEngineCPUUtilization` predefinida é usada para ajustar um cluster ElastiCache (RedisOSS) com base em uma CPU utilização média de 40% em todos os nós primários desse 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": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "ScaleInCooldown": 600,
  "ScaleOutCooldown": 300
}
```

## Desabilitar a atividade de redução

Você pode impedir que a configuração da política de escalabilidade de rastreamento de metas seja escalada em seu cluster desativando a atividade de escalabilidade. 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 do Application Auto Scaling. API

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica `ElastiCachePrimaryEngineCPUUtilization`

predefinida ajusta um OSS cluster ElastiCache com Valkey ou Redis com base em uma CPU utilização média de 40% em todos os nós primários desse 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

Depois de registrar seu cluster ElastiCache com o escalonamento OSS automático Valkey ou Redis e definir uma política de escalabilidade, você aplica a política de escalabilidade ao cluster registrado. Para aplicar uma política de escalabilidade a um cluster ElastiCache (RedisOSS), você pode usar o AWS CLI ou o Application Auto Scaling. API

#### Aplicando uma política de escalabilidade usando o AWS CLI

Para aplicar uma política de escalabilidade ao seu ElastiCache OSS cluster Valkey ou Redis, use o [put-scaling-policy](#) comando 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 esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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 de rastreamento de destino a ser usada no cluster.

No exemplo a seguir, você aplica uma política de escalabilidade de rastreamento de destino nomeada `myscalablepolicy` a um cluster ElastiCache com Valkey ou Redis OSS chamado

com escalabilidade automática. `myscalablecluster` 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
```

Aplicando uma política de escalabilidade usando o API

Para aplicar uma política de escalabilidade ao seu ElastiCache OSS cluster Valkey ou Redis, use o [PutScalingPolicy](#) AWS CLI comando com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id` — O identificador do recurso. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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 de rastreamento de destino a ser usada no cluster.

No exemplo a seguir, você aplica uma política de escalabilidade de rastreamento de destino nomeada `myscalablepolicy` a um cluster ElastiCache com Valkey ou Redis OSS chamado com escalabilidade automática. `myscalablecluster` 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 AWS Management Console, o ou o AWS CLI Application API Auto Scaling.

### Editando uma política de escalabilidade usando o AWS Management Console

Para editar uma política de Auto Scaling para um cluster ElastiCache com Valkey ou Redis OSS

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha 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.

### Editando uma política de escalabilidade usando o e AWS CLI API

Você pode usar o AWS CLI ou o Application Auto Scaling API para editar uma política de escalabilidade da mesma forma que aplica uma política de escalabilidade:

- Ao usar o AWS CLI, especifique o nome da política que você deseja editar no `--policy-name` parâmetro. Especifique novos valores para os parâmetros que você deseja alterar.
- Ao usar o Application Auto Scaling API, especifique o nome da política que você deseja editar no `PolicyName` parâmetro. 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 AWS Management Console, o ou o AWS CLI Application API Auto Scaling.

### Excluindo uma política de escalabilidade usando o AWS Management Console

Para excluir uma política de Auto Scaling para um cluster ElastiCache (RedisOSS)

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon 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).

## Excluindo uma política de escalabilidade usando o AWS CLI

Para excluir uma política de escalabilidade para você ElastiCache com o Valkey ou o OSS cluster Redis, use o [delete-scaling-policy](#) AWS CLI comando com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id` — O identificador do recurso. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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, você exclui uma política de escalabilidade de rastreamento de destino chamada `myscalablepolicy` 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
```

## Excluindo uma política de escalabilidade usando o API

Para excluir uma política de escalabilidade para você ElastiCache com o Valkey ou o OSS cluster Redis, use o [DeleteScalingPolicy](#) AWS CLI comando com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.



- `--resource-id` — O identificador do recurso. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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, você exclui uma política de escalabilidade de rastreamento de destino chamada `myscalablepolicy` 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 AWS CloudFormation para políticas de Auto Scaling

Este trecho mostra como criar uma política de rastreamento de alvos e aplicá-la a um recurso [AWS::ElastiCache::](#) usando o `ReplicationGroup` 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 o escalonamento programado, você cria ações agendadas, que orientam o ElastiCache Valkey ou o Redis OSS a realizar atividades de escalabilidade em horários específicos. Ao criar uma ação agendada, você especifica um cluster existente ElastiCache (RedisOSS), quando a atividade de escalabilidade deve ocorrer, capacidade mínima e capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

Você só pode criar uma ação agendada para clusters ElastiCache (RedisOSS) que já existem. 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 AWS Management Console e abra o ElastiCache console da Amazon 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 AWS Management Console e abra o ElastiCache console da Amazon 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 AWS Management Console e abra o ElastiCache console da Amazon 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 o seguinte escalonamento automático de aplicativos: APIs

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Use AWS CloudFormation para criar uma ação programada

Este trecho mostra como criar uma política de rastreamento de alvos e aplicá-la a um recurso [AWS::ElastiCache::](#) usando o ReplicationGroup 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 ElastiCache de replicação pode configurar um ou mais caches para funcionar como um único nó lógico.

A seguir, são fornecidos detalhes sobre o rastreamento de metas e as políticas programadas e como aplicá-las usando o AWS Management Console AWS CLI APIs e.

### 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. ElastiCache com Valkey ou Redis OSS AutoScaling cria e gerencia os CloudWatch alarmes que acionam a política de escalabilidade e calcula o ajuste de escala com base na métrica e no valor alvo. 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 CPU utilização AVG do mecanismo agregado em todas as réplicas usadas para acionar uma operação de ElastiCache auto-escalamento. Você pode definir a meta de utilização entre 35% e 70%.

Quando o serviço detecta que sua `ElastiCacheReplicaEngineCPUUtilization` métrica é igual ou maior que a configuração do `Target`, ele aumentará automaticamente as réplicas em seus fragmentos. ElastiCache com Valkey ou Redis OSS expande suas réplicas de cluster em uma contagem igual ao maior de dois números: variação percentual do `Target` e uma réplica. Para escalar, ElastiCache com Valkey ou Redis, OSS não haverá escalabilidade automática, a menos que o valor geral da métrica esteja abaixo de 75 por cento da meta definida.

Para um exemplo de aumento de escala na horizontal, se você tiver 5 fragmentos e 1 réplica cada:

Se seu `Target` violar em 30 por cento, ElastiCache com Valkey ou Redis, OSS expanda em 1 réplica (máximo (0,3, padrão 1)) em todos os fragmentos, o que resulta em 5 fragmentos com 2 réplicas cada,

Por exemplo, se você selecionou o valor alvo de 60%, ElastiCache com o Valkey ou o Redis, a escalabilidade OSS não será automática até que a métrica seja menor ou igual a 45% (25% abaixo da meta 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 metas para escalar quando a métrica especificada está abaixo do valor alvo. ElastiCache com Valkey ou Redis OSS expande as réplicas em no máximo (% de desvio arredondado em relação ao `Target`, padrão 1) das réplicas existentes em todos os fragmentos do 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, ElastiCache com o Valkey ou o RedisOSS, o Auto Scaling sempre age de forma conservadora, arredondando para cima ou para baixo ao determinar quanta capacidade adicionar ou remover. 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 de rastreamento de destino para um OSS cluster ElastiCache com Valkey ou Redis, desde que cada uma delas use uma métrica diferente. A intenção do Auto Scaling é sempre priorizar a disponibilidade, portanto, seu comportamento difere dependendo se as políticas de rastreamento de alvos estão prontas para expansão horizontal ou ampliação. 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 CloudWatch alarmes que, ElastiCache com o Valkey ou o Redis OSS Auto Scaling, gerencia para uma política de escalabilidade de rastreamento de destino. O Auto Scaling exclui os alarmes automaticamente quando você exclui a política de escalabilidade ou exclui o cluster.
- ElastiCache com Valkey ou Redis, o Auto OSS Scaling não impede que você modifique manualmente as réplicas entre fragmentos. Esses ajustes manuais não afetam nenhum CloudWatch alarme existente associado à política de escalabilidade, mas podem afetar as métricas que podem acionar esses CloudWatch alarmes.
- Esses CloudWatch alarmes gerenciados pelo Auto Scaling são definidos pela AVG métrica em todos os fragmentos do cluster. Assim, ter fragmentos quentes pode resultar em qualquer cenário de:
  - dimensionamento quando não é necessário devido à carga em alguns fragmentos quentes que acionam um alarme CloudWatch
  - não é escalável quando necessário devido à agregação de AVG todos os fragmentos que afeta o alarme de não violação.
- ElastiCache com Valkey ou Redis, os limites OSS padrão de 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 no seu VPC, o que é necessário durante a expansão. Para obter mais informações, consulte [Interfaces de rede elástica](#).
- Se não houver capacidade suficiente disponível em EC2, ElastiCache com Valkey ou Redis OSS, o Auto Scaling não escalaria até que a capacidade estivesse disponível ou se você modificasse manualmente o cluster para os tipos de instância com capacidade suficiente.

- ElastiCache com Valkey ou Redis O Auto OSS Scaling não oferece suporte ao escalonamento de réplicas com um cluster `ReservedMemoryPercent` com menos de 25%. Para obter mais informações, consulte [Gerenciando a memória reservada para Valkey e Redis OSS](#).

## Adicionar uma política de escalabilidade

Você pode adicionar uma política de escalabilidade usando o AWS Management Console

Adicionar uma política de escalabilidade usando o AWS Management Console

Para adicionar uma política de escalonamento automático ElastiCache com Valkey ou Redis OSS

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon 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 alvo, digite a porcentagem média de CPU utilização que você deseja manter nas ElastiCache réplicas. Esse valor deve ser  $\geq 35$  e  $\leq 70$ . 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 o AWS CLI para modificar os valores de resfriamento.
11. Em Capacidade mínima, digite o número mínimo de réplicas que a política ElastiCache com Valkey ou Redis OSS Auto Scaling precisa manter.
12. Em Capacidade máxima, digite o número máximo de réplicas que a política de OSS Auto Scaling ElastiCache com Valkey ou Redis deve manter. Esse valor deve ser  $\geq 5$ .
13. Escolha Criar.



## Registrar uma meta estável

Você pode aplicar uma política de escalabilidade com base em uma métrica predefinida ou personalizada. Para fazer isso, você pode usar o AWS CLI ou o Application Auto Scaling API. A primeira etapa é registrar seu grupo de OSS replicação Valkey ou Redis ElastiCache com o Auto Scaling.

Antes de usar o escalonamento ElastiCache automático com um cluster, você deve registrar seu cluster ElastiCache com o escalonamento OSS automático Valkey ou Redis. Você faz isso para definir a dimensão e os limites de escalabilidade a serem aplicados a esse cluster. ElastiCache com o Valkey ou o Redis, o OSS auto scaling dimensiona dinamicamente o cluster ao longo da dimensão `elasticache:replication-group:Replicas` escalável, que representa o número de réplicas de cluster por fragmento.

### Usando o CLI

Para registrar seu ElastiCache cluster, use o [register-scalable-target](#) comando com os seguintes parâmetros:

- `--service-namespace`: defina este valor como `elasticache`.
- `--resource-id` — O identificador do recurso para o cluster. ElastiCache Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster, por exemplo `replication-group/myscalablecluster`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:Replicas`.
- `--min-capacity` — O número mínimo de réplicas a serem gerenciadas ElastiCache com o escalonamento automático Valkey ou Redis. OSS 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 réplicas a serem gerenciadas ElastiCache com o escalonamento automático Valkey ou Redis. OSS 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, você registra um OSS cluster ElastiCache com Valkey ou Redis chamado `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 ^  

```

Usando o API

Para registrar seu ElastiCache cluster, use o [register-scalable-target](#) comando com os seguintes parâmetros:

- **ServiceNamespace** — Defina esse valor como `elasticache`.
- **resourceID** — O identificador do recurso para o cluster. ElastiCache Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster, por exemplo `replication-group/myscalablecluster`.
- **ScalableDimension** — Defina esse valor como `elasticache:replication-group:Replicas`.
- **MinCapacity** — O número mínimo de réplicas a serem gerenciadas ElastiCache com o OSS escalonamento automático Valkey ou Redis. 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 ElastiCache com o OSS escalonamento automático Valkey ou Redis. 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, você registra um cluster chamado `myscalablecluster` com o Application Auto Scaling API. 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 ajuste de escala

Uma configuração de política de escalabilidade de rastreamento de metas é representada por um JSON bloco no qual as métricas e os valores-alvo são definidos. Você pode salvar uma configuração de política de escalabilidade como um JSON bloco em um arquivo de texto. Você usa esse arquivo de texto ao invocar o AWS CLI ou o Application API Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de políticas, consulte [TargetTrackingScalingPolicyConfiguration](#) na Application Auto API Scaling Reference.

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 AWS CloudFormation para políticas de Auto Scaling](#)
- [Escalabilidade programada](#)

### Uso de uma métrica predefinida

Uma configuração de política de escalabilidade de rastreamento de metas é representada por um JSON bloco no qual as métricas e os valores-alvo são definidos. Você pode salvar uma configuração de política de escalabilidade como um JSON bloco em um arquivo de texto. Você usa esse arquivo de texto ao invocar o AWS CLI ou o Application API Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de políticas, consulte [TargetTrackingScalingPolicyConfiguration](#) na Application Auto API Scaling Reference.

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 ElastiCache com Valkey ou Redis OSS](#)

### Uso de uma métrica predefinida

Usando métricas predefinidas, você pode definir rapidamente uma política de escalabilidade de rastreamento de metas para um OSS cluster ElastiCache com Valkey ou Redis que funcione com rastreamento de alvos com Valkey ou Redis Auto Scaling. ElastiCache OSS Atualmente, é ElastiCache compatível com a seguinte métrica predefinida no Auto Scaling de ElastiCache réplicas:

`ElastiCacheReplicaEngineCPUUtilization`— O valor médio da `ngineCPUUtilization` métrica E em CloudWatch todas as réplicas no cluster. Você pode encontrar o valor da métrica agregada CloudWatch em `ElastiCache ReplicationGroupId`, `Role for required ReplicationGroupId` e `Role Replica`.

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 OSS métrica ElastiCache com Valkey ou Redis que mude em proporção à escala. Nem todas as ElastiCache métricas funcionam para o rastreamento de metas. 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 CPU utilização média de 50% em todas as réplicas em um cluster chamado `my-db-cluster`

```
{
  "TargetValue": 50,
  "CustomizedMetricSpecification":
  {
    "MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      {
        "Name": "RelicationGroup",
        "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 do Application Auto Scaling. API O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica `ElastiCacheReplicaEngineCPUUtilization` predefinida é usada para ajustar um cluster com base em uma CPU utilização média de 40% em todas as réplicas desse 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 de rastreamento de metas seja escalada em seu cluster ElastiCache com Valkey ou Redis OSS desativando a atividade de escalabilidade. 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 do Application Auto Scaling. API

## Example

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica `ElastiCacheReplicaEngineCPUUtilization` predefinida ajusta um cluster com base em uma CPU utilização média de 40% em todas as réplicas desse cluster. A configuração desativa a atividade de redução para a política de escalabilidade.

```
{"TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"},
}
```

```
"DisableScaleIn": true
}
```

## Aplicação de uma política de escalabilidade a um cluster ElastiCache com Valkey ou Redis OSS

Depois de registrar seu cluster ElastiCache com o escalonamento OSS automático Valkey ou Redis e definir uma política de escalabilidade, você aplica a política de escalabilidade ao cluster registrado. Para aplicar uma política de escalabilidade a um OSS cluster ElastiCache com Valkey ou Redis, você pode usar o AWS CLI ou o Application Auto Scaling. API

### Usando o AWS CLI

Para aplicar uma política de escalabilidade ao seu ElastiCache OSS cluster Valkey ou Redis, use o [put-scaling-policy](#) comando 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 esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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 de rastreamento de destino a ser usada no cluster.

### Example

No exemplo a seguir, você aplica uma política de escalabilidade de rastreamento de metas nomeada `myscalablepolicy` a um cluster chamado `myscalablecluster` ElastiCache com escalabilidade automática Valkey ou Redis. OSS 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
```

```
--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
```

## Usando o API

Para aplicar uma política de escalabilidade ao seu OSS cluster ElastiCache com Valkey ou Redis com o Application Auto API Scaling, use a operação Application Auto [PutScalingPolicy](#) Scaling com os seguintes API parâmetros:

- **PolicyName** — O nome da política de escalabilidade.
- **PolicyType** — Defina esse valor como `TargetTrackingScaling`.
- **resourceID** — O identificador do recurso para o cluster. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster ElastiCache (RedisOSS), por exemplo `replication-group/myscalablecluster`.
- **ServiceNamespace** — Defina esse valor como `elasticache`.
- **ScalableDimension** — Defina esse valor como `elasticache:replication-group:Replicas`.
- **TargetTrackingScalingPolicyConfiguration** — A configuração da política de escalabilidade de rastreamento de metas a ser usada no cluster.



## Example

No exemplo a seguir, você aplica uma política de escalabilidade de rastreamento de metas nomeada `scalablepolicy` a um cluster nomeado `myscalablecluster` ElastiCache com escalabilidade automática 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 AWS Management Console, o ou o AWS CLI Application API Auto Scaling.

## Editando uma política de escalabilidade usando o AWS Management Console

Você só pode editar políticas com métricas do tipo Predefined (Predefinidas) usando o AWS Management Console

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon 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.

Editando uma política de escalabilidade usando o AWS CLI ou o Application Auto Scaling API

Você pode usar o AWS CLI ou o Application Auto Scaling API para editar uma política de escalabilidade da mesma forma que aplica uma política de escalabilidade:

- Ao usar o Application Auto Scaling API, especifique o nome da política que você deseja editar no `PolicyName` parâmetro. Especifique novos valores para os parâmetros que você deseja alterar.

Para obter mais informações, consulte [Aplicação de uma política de escalabilidade a um cluster ElastiCache com Valkey ou Redis OSS](#).

Excluir uma política de escalabilidade

Você pode excluir uma política de escalabilidade usando o AWS Management Console, o AWS CLI ou o Application Auto Scaling API

Excluindo uma política de escalabilidade usando o AWS Management Console

Você só pode editar políticas com métricas do tipo Predefined (Predefinidas) usando o AWS Management Console

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon 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).

Excluindo uma política de escalabilidade usando o AWS CLI ou o Application Auto Scaling API

Você pode usar o AWS CLI ou o Application Auto Scaling API para excluir uma política de escalabilidade de um cluster. ElastiCache

## CLI

Para excluir uma política de escalabilidade do seu OSS cluster ElastiCache com Valkey ou Redis, use o [delete-scaling-policy](#) comando com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id` — O identificador do recurso para o cluster. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o 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, você exclui uma política de escalabilidade de rastreamento de destino chamada `myscalablepolicy` de um cluster chamado `ELC`; `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 \  
  --
```

## 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 OSS cluster ElastiCache com Valkey ou Redis, use a operação Application Auto [DeleteScalingPolicy](#) API Scaling com os seguintes parâmetros:

- **PolicyName** — O nome da política de escalabilidade.
- **resourceID** — O identificador do recurso para o cluster. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster, por exemplo `replication-group/myscalablecluster`.
- **ServiceNamespace** — Defina esse valor como `elasticache`.
- **ScalableDimension** — Defina esse valor como `elasticache:replication-group:Replicas`.

No exemplo a seguir, você exclui uma política de escalabilidade de rastreamento de destino chamada `myscalablepolicy` de um cluster chamado `myscalablecluster` com o Application Auto Scaling. API

```
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 AWS CloudFormation para políticas de Auto Scaling

Este trecho mostra como criar uma ação agendada e aplicá-la a um recurso [AWS::ElastiCache:ReplicationGroup](#) usando o [AWS::ApplicationAutoScaling:ScalableTarget](#) 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 o escalonamento programado, você cria ações agendadas, que orientam o ElastiCache Valkey ou o Redis OSS a realizar atividades de escalabilidade em horários específicos. Ao criar uma ação agendada, você especifica um OSS cluster existente ElastiCache com Valkey ou Redis, quando a atividade de escalabilidade deve

ocorrer, capacidade mínima e capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

Você só pode criar uma ação ElastiCache agendada para OSS clusters Valkey ou Redis que já existem. 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 o seguinte escalonamento automático de aplicativos: APIs

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Use AWS CloudFormation para criar políticas de Auto Scaling

Este trecho mostra como criar uma ação agendada e aplicá-la a um recurso [AWS::ElastiCache:](#) usando o ReplicationGroup 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 distribuídos na memória que oferecem suporte à fragmentação e à replicação. ElastiCache Os OSS clusters Valkey e Redis são a implementação distribuída que permite que os dados sejam particionados em vários nós. Um cluster ElastiCache (RedisOSS) tem dois modos de operação, o modo de cluster ativado (CME) e o modo de cluster desativado (CMD). NoCME, um OSS mecanismo Valkey e Redis funciona como um banco de dados distribuído com vários fragmentos e nós, enquanto noCMD, o Valkey e o Redis OSS funcionam como um único nó.

Antes de migrar de CMD paraCME, 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.
- Os aplicativos devem usar um OSS cliente Valkey ou Redis que seja capaz de usar o protocolo Cluster e usar 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 Valkey 7.2 e superior ou Redis OSS 7.0 e superior.

Para migrar de CMD para CME, a configuração do modo de cluster deve ser alterada do modo de cluster desativado para o modo de 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 (RedisOSS) 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. 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 ter mais informações, consulte [Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros](#)

1. Prepare-se — Crie um CME cluster de teste e certifique-se de que sua pilha esteja pronta para trabalhar com ele. ElastiCache (RedisOSS) não tem como verificar se você está pronto. Para obter mais informações, consulte [Criação de um cluster para Valkey ou Redis OSS](#).
2. Modifique a configuração de CMD cluster existente para ser compatível com o modo de cluster — Nesse modo, haverá um único fragmento implantado e ElastiCache (RedisOSS) 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, os aplicativos devem ser reconfigurados para começar a usar o protocolo Valkey ou o Redis OSS Cluster e o endpoint de configuração. Para alterar o modo de OSS cluster Valkey ou Redis para o modo de 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 (excluindo `cacheParameterGroupName`) não podem ser modificados ao definir o parâmetro do modo de cluster na solicitação. [ModifyReplicationGroup](#)



- a. Usando o AWS Management Console, veja [Modificação de um grupo de replicação](#) e defina o modo de cluster como Compatível
- b. Usando o API, veja [ModifyReplicationGroup](#) atualize o `ClusterMode` parâmetro para `compatible`.
- c. Usando o AWS CLI, veja [modify-replication-group](#) atualize o `cluster-mode` parâmetro para `compatible`.

Depois de alterar o modo de OSS cluster Valkey ou Redis para o modo de cluster compatível, eles [DescribeReplicationGroups](#) API retornarão o endpoint de configuração do cluster ElastiCache (RedisOSS). 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 [Encontrando pontos de extremidade de conexão em 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, verifique se seus OSS clientes Valkey ou Redis migraram para o uso do protocolo de cluster e se o endpoint de configuração do cluster não está em uso.

- a. Usando o AWS Management Console, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Ativado.
- b. Usando o API, veja [ModifyReplicationGroup](#) atualize o `ClusterMode` parâmetro para `enabled`.
- c. Usando o AWS CLI, veja [modify-replication-group](#) atualize o `cluster-mode` parâmetro para `enabled`.

Depois de alterar o modo de cluster para ativado, os endpoints serão configurados de acordo com a especificação do cluster Valkey ou RedisOSS. O [DescribeReplicationGroups](#) API retornará o parâmetro do modo de cluster `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 para o modo de cluster desativado (CMD) do 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 AWS Management Console, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Desativado
2. Usando o API, veja [ModifyReplicationGroup](#) atualize o `ClusterMode` parâmetro `paradisabled`.
3. Usando o AWS CLI, veja [modify-replication-group](#) atualize o `cluster-mode` parâmetro `paradisabled`.

Depois de alterar o modo de cluster para desativado, o [DescribeReplicationGroups](#) API 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 autoprojetoados.

Ao usar o recurso Global Datastore, você pode trabalhar com replicação de OSS cluster Valkey ou Redis 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](#)
- [Pré-requisitos e limitações](#)

- [Uso de datastores globais \(console\)](#)
- [Usando armazenamentos de dados globais \(\) CLI](#)

## Visão geral

Cada datastore global é uma coleção de um ou mais clusters que são replicados entre si.

Um datastore global consiste no seguinte:

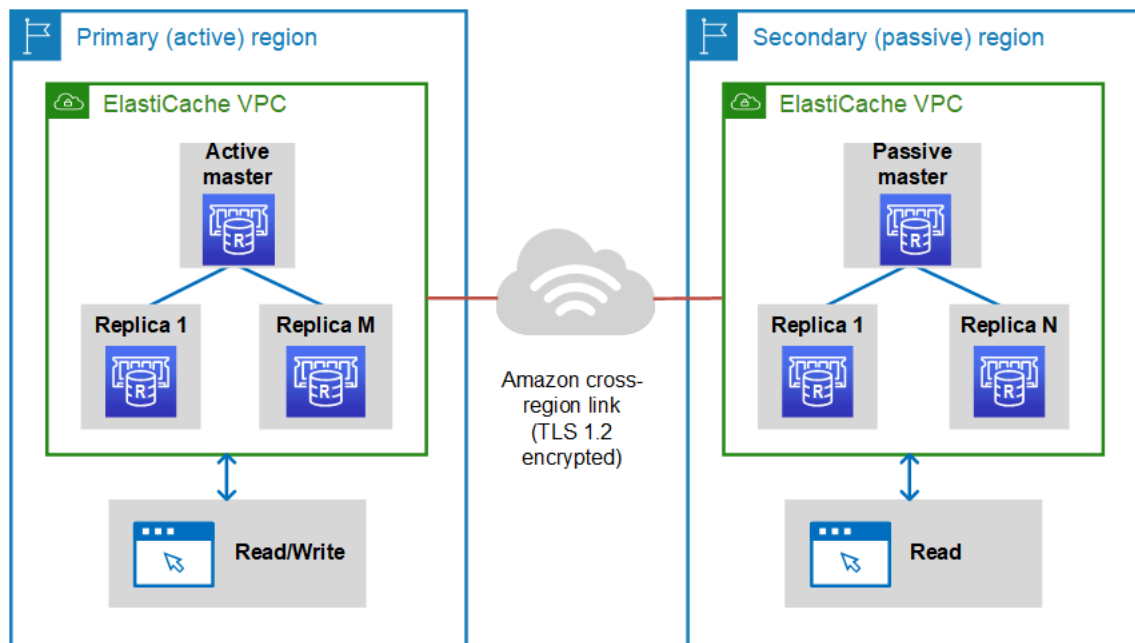
- **Cluster primário (ativo):** um cluster primário aceita gravações que são replicadas para todos os clusters dentro do datastore global. Um cluster primário também aceita solicitações de leitura.
- **Cluster secundário (passivo):** um cluster secundário só aceita solicitações de leitura e replica as 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 com o Valkey ou o RedisOSS, ele replica automaticamente seus dados do cluster primário para o cluster secundário. Você escolhe a AWS região em que os OSS dados do Valkey ou do Redis 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.

Usar um armazenamento de dados 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 armazenamentos de dados globais são suportados nas seguintes AWS regiões: Ásia-Pacífico (Seul, Tóquio, Cingapura, Sydney, Mumbai e Osaka), Europa (Frankfurt, Paris, Londres, Irlanda e Estocolmo), Leste dos EUA (Norte da Virgínia e Ohio), Oeste dos EUA (Norte da Califórnia e Oregon), América do Sul (São Paulo), (Oeste dos EUA e Leste dos EUA), Região do Canadá AWS GovCloud (Central), China (Pequim e Ningxia)
- 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 armazenamentos de dados globais são compatíveis com instâncias de tamanho grande ou superior.
- Você pode configurar a replicação de um cluster primário de uma AWS região para um cluster secundário em até duas outras AWS regiões.

**Note**

A exceção a isso são regiões China (Pequim) e China (Ningxia), onde a replicação só pode ocorrer entre as duas regiões.

- Você pode trabalhar com datastores globais somente em VPC clusters. Para obter mais informações, consulte [Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC](#). Não há suporte para armazenamentos de dados globais quando você usa EC2 -Classic. Para obter mais informações, consulte [EC2-Classic](#) no Guia do EC2 usuário da Amazon.

**Note**

No momento, você não pode usar datastores globais no [Usando zonas locais com ElastiCache](#).

- ElastiCache não suporta failover automático de uma AWS região para outra. Quando necessário, você pode promover um cluster secundário manualmente. Para ver 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](#), [criptografia em trânsito](#) e [AUTH](#).
- Os armazenamentos de dados globais não oferecem suporte ao Internet Protocol versão 6 (IPv6).
- Os armazenamentos de dados globais oferecem suporte a chaves AWS KMS . 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 AWS região primária são propagados para as AWS regiões secundárias.
- No modo de cluster habilitado, o seguinte se aplica:
  - Para eventos publicados que não estão em um keypace, somente assinantes da mesma AWS região recebem os eventos.
  - Para eventos publicados no keypace, os assinantes de todas as AWS regiões recebem os 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 Valkey 7.2 ou posterior ou Redis OSS 5.0.6 ou posterior.
2. Adicione até dois clusters secundários em AWS regiões diferentes, novamente usando o Valkey 7.2 ou posterior, ou o mecanismo Redis OSS 5.0.6 ou posterior.

Os procedimentos a seguir orientam você sobre como criar um armazenamento de dados global para Valkey ou Redis OSS e realizar outras operações usando o console. 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 que seja Valkey 7.2 ou posterior ou Redis OSS 5.0.6 ou posterior.

## Como criar um datastore global usando um cluster existente

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados globais e, em seguida, escolha Criar armazenamento de dados global.
3. Na página Configurações do cluster primário, faça o seguinte:
  - No campo Informações do armazenamento de dados global, insira um nome para o novo armazenamento de dados 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**

ElastiCache gera automaticamente um novo grupo de parâmetros a partir dos 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**

Você pode fornecer uma chave de criptografia diferente escolhendo AWS KMSChave gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando AWS KMS chaves 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 Valkey 7.2 e versões posteriores e Redis OSS Engine 6.0 em diante, se você habilitar a criptografia em trânsito, será solicitado que especifique 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 [Gerenciando grupos de usuários com o console e CLI](#).
    - AUTHUsuário padrão — Um mecanismo de autenticação para um servidor Valkey ou RedisOSS. 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.
- Port (Porta)
  - Número de réplicas



- Grupo de sub-redes
  - Zona(s) de disponibilidade preferencial(is)
  - Grupos de segurança
  - Gerenciado pelo cliente (AWS KMSChave)
  - AUTHToken
  - Habilitar backups automáticos
  - Backup retention period (Período de retenção de backup)
  - Janela de backup
  - Janela de manutenção
  - Tópico para SNS notificação
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 armazenamento de dados 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 adicional em uma AWS região 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 AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados globais e, em seguida, escolha Criar armazenamento de dados 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. Para obter informações do Global Datastore, insira um valor para Nome. ElastiCache usa o sufixo para gerar um nome exclusivo para o armazenamento de dados 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 Região, escolha uma AWS região 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 Valkey e Redis OSS](#) e [Criação de um grupo de ElastiCache parâmetros](#).

 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 [Hierarquização de dados em ElastiCache](#).

- e. Se você estiver criando um cluster Valkey ou Redis OSS (modo de cluster desativado):

Em Number of replicas (Número de réplicas), escolha o número de réplicas desejado para esse cluster.

- f. Se você estiver criando um cluster Valkey ou Redis OSS (modo de cluster ativado):

- i. Em Número de fragmentos, escolha o número de fragmentos (partições/grupos de nós) que você deseja para esse cluster Valkey ou Redis OSS (modo de cluster ativado).

Para algumas versões do Valkey ou do Redis OSS (modo de cluster ativado), você pode alterar dinamicamente o número de fragmentos em seu cluster:

- Redis OSS 3.2.10 e versões posteriores — Se seu cluster estiver executando o Redis OSS 3.2.10 ou versões posteriores, você poderá alterar dinamicamente o número de fragmentos em seu cluster. Para obter mais informações, consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#).
- Outras OSS versões do Redis — Se seu 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 Valkey ou Redis OSS (modo de cluster ativado).

- 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: semeando um novo cluster autoprojetoado com um backup criado externamente](#).
6. Em Configurações de grupo de sub-rede, escolha a sub-rede que você deseja aplicar a esse cluster. 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 IPv6 CIDR bloco. Se você escolher pilha dupla, deverá selecionar um tipo de IP de descoberta, IPv6 ou IPv4.

Para obter mais informações, consulte [Criar uma sub-rede no seu VPC](#).

7. 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 [Escolhendo regiões e zonas de disponibilidade para 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 a AWS KMSchave gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando chaves gerenciadas pelo cliente de 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 superior e o OSS mecanismo Redis versão 6.0 e superior, se você ativar a criptografia em trânsito, será solicitado que especifique 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 [Gerenciando grupos de usuários com o console e CLI](#).
- AUTHUsuário padrão — Um mecanismo de autenticação para um servidor Valkey ou RedisOSS. Para obter mais informações, consulte [AUTH](#).
- AUTH— Um mecanismo de autenticação para um servidor Valkey ou RedisOSS. Para obter mais informações, consulte [AUTH](#).

 Note

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

- 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. Você pode usar o grupo de segurança padrão para o seu VPC ou criar um novo.

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

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 janela de manutenção é o horário, geralmente de uma hora de duração, a cada semana em que ElastiCache agenda a manutenção do sistema para seu cluster. Você pode ElastiCache escolher o dia e a hora da janela de manutenção (sem preferência), ou você mesmo pode escolher o dia, a hora e a duração (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 tempos são UCT tempos.

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

12. (Opcional) Para Logs:

- Em Formato de registro, escolha 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,

13. 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](#).

14. Revise todas as suas entradas e opções e faça as correções necessárias. Quando estiver pronto, escolha Next (Avançar).


15. Depois de configurar o cluster nas etapas anteriores, agora você vai configurar os detalhes do cluster secundário.

16. Em Cluster regional, escolha a AWS região em que 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

ElastiCache gera automaticamente um novo grupo de parâmetros a partir dos 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: permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

 Note

Você pode fornecer uma chave de criptografia diferente escolhendo `AWS KMSChave` gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando AWS KMS chaves 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 superior e o OSS mecanismo Redis versão 6.4 e superior, se você ativar a criptografia em trânsito, será solicitado que especifique 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 [Gerenciando grupos de usuários com o console e CLI](#).
  - AUTHUsuário padrão — Um mecanismo de autenticação para um servidor Valkey ou RedisOSS. Para obter mais informações, consulte [AUTH](#).

 Note

Para OSS versões do Redis entre 4.0.2, quando a criptografia em trânsito foi suportada pela primeira vez, e 6.0.4, 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:



- Port (Porta)
- Número de réplicas
- Grupo de sub-redes
- Zona(s) de disponibilidade preferencial(is)
- Grupos de segurança
- Gerenciado pelo cliente (AWS KMSChave)
- AUTHToken
- Habilitar backups automáticos
- Backup retention period (Período de retenção de backup)
- Janela de backup
- Janela de manutenção
- Tópico para SNS notificação

19. Escolha Criar. 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 armazenamento de dados 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 adicional em uma AWS região diferente. Para obter mais informações, consulte [Adição de uma região a um datastore global](#).

### Visualização de detalhes do datastore global

Você pode ver os detalhes dos armazenamentos de dados globais existentes e também modificá-los na página Armazenamentos de dados globais.

## Como visualizar os detalhes do datastore global

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados globais e, em seguida, escolha um armazenamento de dados 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
  - Modifying
  - 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 OSS mecanismo Valkey ou Redis que executa o armazenamento de dados 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: Ativado ou desativado

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:

- Região - A AWS região 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 autônomo nessa AWS região.
  - Disassociated (Desassociado): o cluster secundário foi removido do datastore global e agora é um cluster independente em sua região da AWS .
- Atraso na réplica do armazenamento de dados global — mostra um valor por AWS região secundária no armazenamento de dados global. Esse é o atraso entre o nó primário da região secundária e o nó primário da região primária. Para Valkey ou Redis habilitado para o modo de clusterOSS, o atraso indica o atraso máximo em segundos entre os fragmentos.

### Adição de uma região a um datastore global


Você pode adicionar até uma AWS região adicional a um armazenamento de dados global existente. Nesse cenário, você está criando um cluster somente para leitura em uma AWS região separada que recebe atualizações automáticas e assíncronas do cluster primário.

Para adicionar uma AWS região a um armazenamento de dados global

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados globais e selecione um armazenamento de dados global existente.
3. Escolha Adicionar cluster regional e escolha a AWS região em que o cluster secundário residirá.
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

ElastiCache gera automaticamente um novo grupo de parâmetros a partir dos 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

Você pode fornecer uma chave de criptografia diferente escolhendo `AWS KMSChave` gerenciada pelo cliente e escolhendo 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:

- Port (Porta)
- Número de réplicas
- Grupo de sub-redes
- Zona(s) de disponibilidade preferencial(is)
- Grupos de segurança

- AUTHToken
- Habilitar backups automáticos
- Backup retention period (Período de retenção de backup)
- Janela de backup
- Janela de manutenção
- Tópico para SNS notificação

## 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 AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados globais e, em Nome do armazenamento de dados global, escolha um armazenamento de dados global.
3. Escolha Modify (Modificar) e escolha entre as seguintes opções:
  - Modify description (Modificar descrição): atualiza a descrição do datastore global
  - Modificar a versão do mecanismo — Somente o Valkey 7.2 e posterior ou o OSS mecanismo Redis versão 5.0.6 e posterior 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 ativa o failover e os nós primários em clusters regionais são desligados inesperadamente, efetua ElastiCache o failover para uma das réplicas regionais. Para obter mais informações, consulte [Failover automático](#).

Para clusters Valkey ou Redis com o modo de OSS cluster ativado:

- Add shards (Adicionar fragmentos): insira o número de fragmentos a serem adicionados e, opcionalmente, especifique uma ou mais zonas de disponibilidade.
- Excluir fragmentos — Escolha fragmentos a serem excluídos em cada AWS região.
- 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 armazenamento de dados global, modifique o grupo de parâmetros de qualquer cluster membro do armazenamento de dados global. ElastiCache aplica essa alteração a todos os clusters dentro desse armazenamento de dados global automaticamente. Para modificar o grupo de parâmetros desse cluster, use o OSS console Valkey ou Redis ou a [ModifyCacheCluster](#) API operação. Para obter mais informações, consulte [Modificando um grupo de ElastiCache parâmetros](#). 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 [ResetCacheParameterGroup](#) API operação.

Promoção do cluster secundário para primário

Se o cluster ou AWS região principal ficar indisponível ou estiver enfrentando problemas de desempenho, 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 promove vários clusters secundários simultaneamente, ElastiCache com Valkey ou Redis, OSS não garante qual deles, em última análise, será resolvido como primário.

Como promover um cluster secundário para primário

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados 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 AWS região de um armazenamento de dados global usando o procedimento a seguir.

Para remover uma AWS região de um armazenamento de dados global

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados 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 AWS região 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 AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Armazenamentos de dados 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).

### Usando armazenamentos de dados globais () CLI

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 o AWS CLI para operações ad hoc (únicas).

### Baixando e configurando o AWS CLI

AWS CLI É executado em Windows, macOS ou Linux. Use o procedimento a seguir para fazer download e configurá-la.



## Para baixar, instalar e configurar o CLI

1. Faça o download da AWS CLI página da web na [interface da linha de AWS comando](#).
2. Siga as instruções para instalar AWS CLI e configurar o AWS CLI no Guia do AWS Command Line Interface usuário.

## Usando o AWS CLI com datastores globais

Use as seguintes CLI operações para trabalhar com armazenamentos de dados 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
```

A Amazon aplica ElastiCache automaticamente um prefixo ao ID global do armazenamento de dados quando ele é criado. Cada AWS região 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 AWS região e seu prefixo de ID de armazenamento de dados global.

| Nome da região/região                                   | Prefixo |
|---------------------------------------------------------|---------|
| Região Leste dos EUA (Ohio)<br>us-east-2                | fpkhr   |
| Região Leste dos EUA (N. da Virgínia)<br>us-east-1      | ldgnf   |
| Região Oeste dos EUA (Norte da Califórnia)<br>us-west-1 | virxk   |

| Nome da região/região                              | Prefixo |
|----------------------------------------------------|---------|
| Região Oeste dos EUA (Oregon)<br>us-west-2         | sgau1   |
| Região Canadá (Central)<br>ca-central-1            | bxodz   |
| Região Ásia-Pacífico (Mumbai)<br>ap-south-1        | erpgt   |
| Região Ásia-Pacífico (Tóquio)<br>ap-northeast-1    | quwsw   |
| Região Ásia-Pacífico (Seul)<br>ap-northeast-2      | lfqnh   |
| Região Ásia-Pacífico (Osaka)<br>ap-northeast-3     | n1apn   |
| Região Ásia-Pacífico (Singapura)<br>ap-southeast-1 | v1qxn   |
| Asia Pacific (Sydney) Region<br>ap-southeast-2     | vbgxd   |
| Região Europa (Frankfurt)<br>eu-central-1          | iudkw   |
| Região Europa (Irlanda)<br>eu-west-1               | gxeiz   |

| Nome da região/região                          | Prefixo |
|------------------------------------------------|---------|
| Região Europa (Londres)<br>eu-west-2           | okuqm   |
| Região Europa (Paris)<br>eu-west-3             | fgjhi   |
| Região América do Sul (São Paulo)<br>sa-east-1 | juxlw   |
| Região da China (Pequim)<br>cn-north-1         | emvgo   |
| Região da China (Ningxia)<br>cn-northwest-1    | ckbem   |
| Região Ásia-Pacífico (Hong Kong)<br>ap-east-1  | knjmp   |
| AWS GovCloud (Oeste dos EUA)<br>us-gov-west-1  | sgwui   |

- [create-replication-group](#)— use essa operação para criar clusters secundários para um armazenamento de dados global fornecendo o nome do armazenamento de dados 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 essa operação e passar um `--global-replication-group-id` valor, ElastiCache inferirá 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 cruzada de Redis para OSS Valkey para ElastiCache GlobalDataStore

Você pode atualizar um grupo de replicação OSS global existente do Redis para o mecanismo Valkey usando o Console ou API CLI

Se você tiver um grupo de replicação OSS global do Redis existente, poderá fazer o upgrade para o Valkey especificando o novo mecanismo e a versão do mecanismo com `modify-global-replication-group` API

Para Linux, macOS ou Unix:

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id myGlobalReplGroup \  
  --engine valkey \  
  --apply-immediately \  
  --engine-version 7.2
```

Para Windows:

```
aws elasticache modify-global-replication-group ^  
  --global-replication-group-id myGlobalReplGroup ^  
  --engine valkey ^  
  --apply-immediately ^  
  --engine-version 7.2
```

Se você tiver um grupo de parâmetros de cache personalizado aplicado ao grupo de replicação OSS global existente do Redis que deseja atualizar, também precisará passar um grupo de parâmetros de cache Valkey personalizado na solicitação. O grupo de parâmetros personalizados Valkey de entrada deve ter os mesmos valores de parâmetros OSS estáticos do Redis que o grupo de parâmetros OSS personalizados do Redis existente.

Para Linux, macOS ou Unix:

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id myGlobalReplGroup \  
  --engine valkey ^  
  --apply-immediately ^  
  --engine-version 7.2
```

```
--engine valkey \  
--engine-version 7.2 \  
--apply-immediately \  
--cache-parameter-group-name myParamGroup
```

Para Windows:

```
aws elasticache modify-global-replication-group ^  
  --global-replication-group-id myGlobalReplGroup ^  
  --engine valkey ^  
  --engine-version 7.2 ^  
  --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-grupo de replicação](#)

```
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 ElastiCache com Valkey ou RedisOSS.

```
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 OSS clusters Amazon ElastiCache Valkey e Redis de nó único são entidades na memória com serviços limitados de proteção de dados (). AOF Se o seu cluster falhar por qualquer motivo, você perderá todos os dados do cluster. No entanto, se você estiver executando um OSS mecanismo Valkey ou Rediss, poderá agrupar de 2 a 6 nós em um cluster com réplicas em que 1 a 5 nós somente para 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.

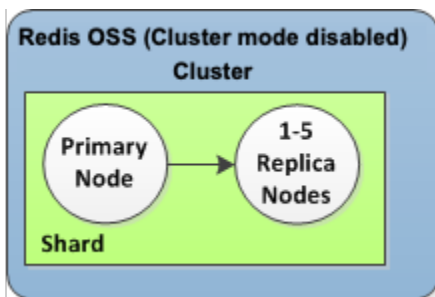
Conforme visto no gráfico a seguir, a estrutura de replicação está contida em um fragmento (chamado grupo de nós noAPI/CLI) que está contido em um cluster Valkey ou Redis. OSS Os clusters Valkey ou Redis OSS (modo de cluster desativado) sempre têm um fragmento. Os clusters Valkey ou Redis OSS (modo de cluster ativado) podem ter até 500 fragmentos com os dados do

cluster particionados entre os 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 nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster com o Valkey e com a versão 5.0.6 ou superior OSS do mecanismo Redis. 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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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).



O cluster Valkey ou Redis OSS (modo de cluster desativado) tem um fragmento e 0 a 5 nós de réplica

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 [Mitigando falhas ao executar Valkey ou Redis OSS](#).

## Tópicos

- [Entendendo a replicação do Valkey e do Redis OSS](#)
- [Replicação: modo de OSS cluster Valkey e Redis desativado versus ativado](#)
- [Minimizando o tempo de inatividade ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#)



- [Como a sincronização e o backup são implementados](#)
- [Criação de um grupo de replicação 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](#)
- [Promovendo uma réplica de leitura para primária, para grupos de replicação Valkey ou Redis OSS \(modo de cluster desativado\)](#)

## Entendendo a replicação do Valkey e do Redis OSS

O Redis OSS implementa a replicação de duas maneiras:

- Com um único fragmento que contém todos os dados do cluster em cada nó — Valkey ou OSS Redis (modo de cluster desativado)
- Com dados particionados em até 500 fragmentos — Valkey ou OSS Redis (modo de cluster ativado)

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 nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do OSS mecanismo Redis 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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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).

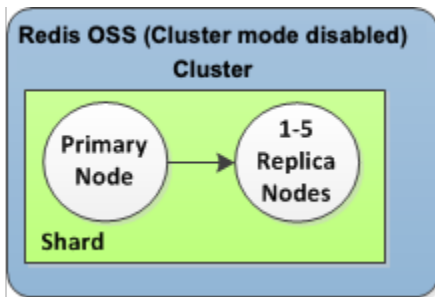
### Tópicos

- [Valkey ou Redis OSS \(modo de cluster desativado\)](#)
- [Valkey ou Redis OSS \(modo de cluster ativado\)](#)

### Valkey ou Redis OSS (modo de cluster desativado)

Um cluster Valkey ou Redis OSS (modo de cluster desativado) tem um único fragmento, dentro do qual há uma coleção de nós; um nó primário de leitura/gravação e até cinco nós de réplica secundários, somente para 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 Valkey ou Redis OSS (modo de cluster desativado) com um único fragmento e nós de réplica

Você pode usar clusters Valkey ou Redis OSS (modo de cluster desativado) com nós de réplica para escalar sua solução ElastiCache para lidar com aplicativos que exigem muita leitura ou para oferecer suporte a um grande número de clientes que leem simultaneamente do mesmo cluster.

Todos os nós em um cluster Valkey ou Redis OSS (modo de cluster desativado) 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 inatividade de gravação, habilite o Multi-AZ com failover automático para seu cluster Valkey ou Redis OSS (modo de cluster desativado) com réplicas. Para obter mais informações, consulte [Minimizando o tempo de inatividade ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#).

Você pode alterar as funções dos nós no cluster Valkey ou Redis OSS (modo de cluster desativado), com o principal 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 [Promovendo uma réplica de leitura para primária, para grupos de replicação Valkey ou Redis OSS \(modo de cluster desativado\)](#).

Valkey ou Redis OSS (modo de cluster ativado)

Um cluster Valkey ou Redis OSS (modo de cluster ativado) 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.

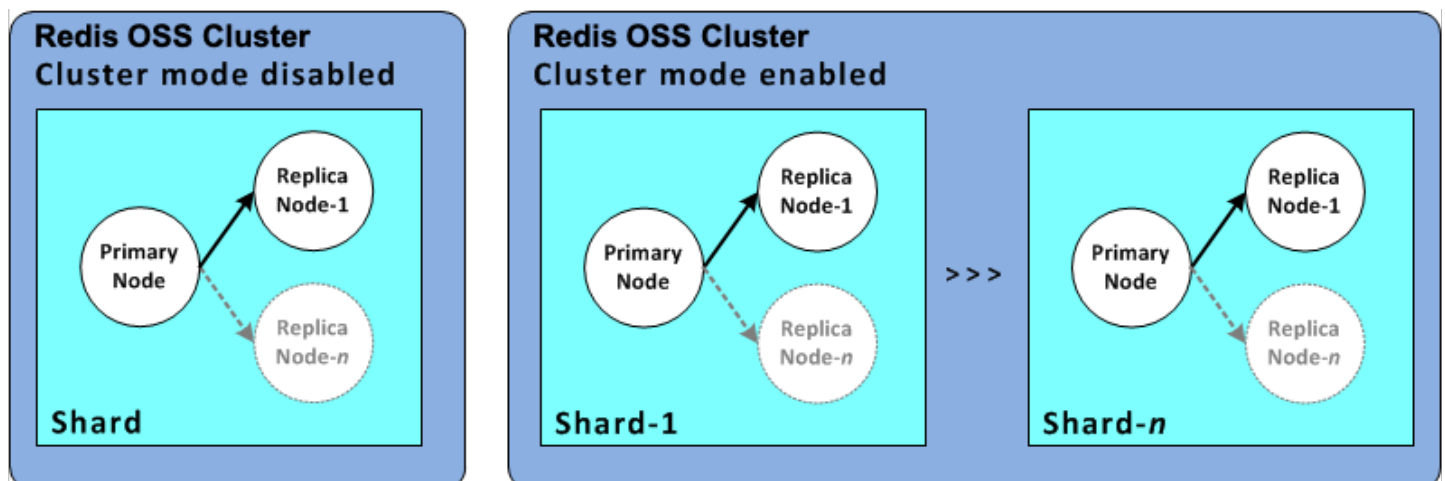
O limite de nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo for Valkey 7.2 e superior ou Redis OSS 5.0.6 e superior. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único e sem réplicas). Verifique se existem endereços IP disponíveis suficientes para acomodar o aumento. As armadilhas comuns incluem que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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 entre os fragmentos em um cluster Valkey ou Redis OSS (modo de cluster ativado).

Os aplicativos usam o endpoint de configuração do cluster Valkey ou Redis OSS (modo de cluster ativado) para se conectar aos nós no cluster. Para obter mais informações, consulte [Encontrando pontos de extremidade de conexão em ElastiCache](#).



Cluster Valkey ou Redis OSS (modo de cluster ativado) com vários fragmentos e nós de réplica

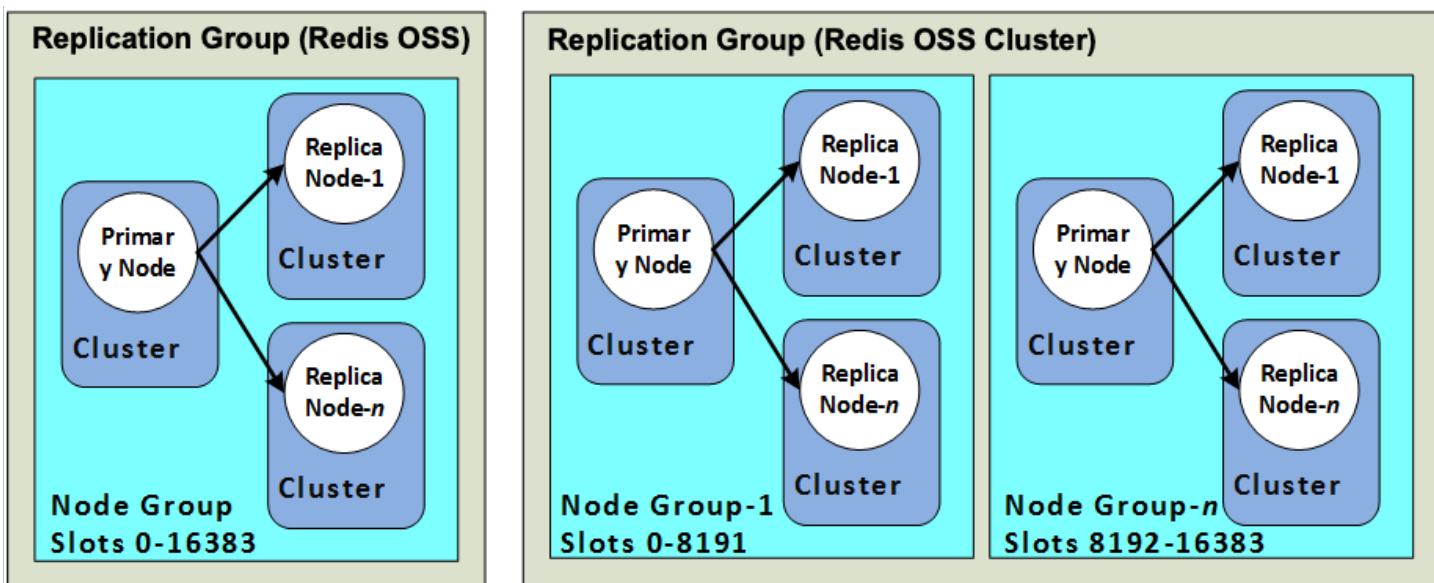
Todos os nós em um cluster Valkey ou Redis OSS (modo de cluster ativado) devem residir na mesma região. Para melhorar a tolerância a falhas, você pode provisionar primários e réplicas de leitura em várias zonas de disponibilidade dentro dessa região.

Atualmente, os recursos Valkey ou Redis OSS (modo de cluster ativado) têm algumas limitações.

- Você não pode promover manualmente nenhum nó de réplica para primário.

## Replicação: modo de OSS cluster Valkey e Redis desativado versus ativado



A partir do Valkey 7.2 e do Redis OSS versão 3.2, você tem a capacidade de criar um dos dois tipos distintos de clusters (API/CLI: grupos de replicação). Um cluster Valkey ou Redis OSS (modo de cluster desativado) sempre tem um único fragmento (API/CLI: grupo de nós) com até 5 nós de réplica de leitura. Um cluster Valkey ou Redis OSS (modo de cluster ativado) tem até 500 fragmentos com 1 a 5 nós de réplica de leitura em cada um.



Clusters Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado)

A tabela a seguir resume as diferenças importantes entre os clusters Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado).

## Comparando clusters Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado)

| Atributo                 | Valkey ou Redis OSS (modo de cluster desativado)                                                                                                                                                  | Valkey ou Redis OSS (modo de cluster ativado)                                                                                                                                                                                          |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modificável              | Sim. Oferece suporte para adicionar e excluir nós de réplica e dimensionar o tipo de nó.                                                                                                          | Limitado. Para ter mais informações, consulte <a href="#">Gerenciamento de versões para ElastiCache</a> e <a href="#">Dimensionamento de clusters em Valkey ou Redis OSS (modo de cluster ativado)</a> .                               |
| Particionamento de dados | Não                                                                                                                                                                                               | Sim                                                                                                                                                                                                                                    |
| Estilhaços               | 1                                                                                                                                                                                                 | 1 a 500                                                                                                                                                                                                                                |
| Réplicas de leitura      | 0 – 5<br><br> <b>Important</b><br>Se você não tiver réplicas e o nó falhar, ocorrerá uma perda total de dados. | 0 a 5 por fragmento.<br><br> <b>Important</b><br>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.<br><br>Opcional. Ativado por padrão.                                                                                                                               | Sim<br><br>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                                                                                                                                                    | Sim, usando arquivos.rdb de um cluster Valkey ou Redis OSS (modo de cluster                                                                                                                                                            |

| Atributo                | Valkey ou Redis OSS (modo de cluster desativado)                                                                                             | Valkey ou Redis OSS (modo de cluster ativado)                                                                                                |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|                         | Valkey ou Redis OSS (modo de cluster desativado).                                                                                            | desativado) ou Valkey ou Redis OSS (modo de cluster ativado).                                                                                |
| Com suporte por         | Todas as versões 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 <a href="#">Gerenciamento de versões para ElastiCache</a> .                   | Sim, com alguns limites. Para obter mais informações, consulte <a href="#">Gerenciamento de versões para ElastiCache</a> .                   |
| Criptografia            | Versões 3.2.6 (programa das paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e 4.0.10 e posteriores. | Versões 3.2.6 (programa das paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e 4.0.10 e posteriores. |
| HIPAAElegível           | Versões 3.2.6 (programa das paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e 4.0.10 e posteriores. | Versões 3.2.6 (programa das paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e 4.0.10 e posteriores. |
| PCIDSSCompatível        | Versões 3.2.6 (programa das paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e 4.0.10 e posteriores. | Versões 3.2.6 (programa das paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e 4.0.10 e posteriores. |
| Reestilhaçamento online | N/D                                                                                                                                          | Versão 3.2.10 (programa da paraEOL, consulte o <a href="#">cronograma de fim de vida das OSS versões do Redis</a> ) e posterior.             |

## Qual devo escolher?

Ao escolher entre Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado), 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 o Redis OSS (modo de cluster desativado) oferecem suporte ao escalonamento. 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 [Dimensionamento de nós de réplica para Valkey ou Redis OSS \(modo de cluster desativado\)](#).

O Valkey ou Redis OSS (modo de cluster ativado) suporta o particionamento de 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 [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#).

- Tamanho do nó versus número de nós — Como um cluster Valkey ou Redis OSS (modo de cluster desativado) tem apenas um fragmento, o tipo de nó deve ser grande o suficiente para acomodar todos os dados do cluster, além da sobrecarga necessária. Por outro lado, como você pode particionar seus dados em vários fragmentos ao usar um cluster Valkey ou Redis OSS (modo de cluster ativado), 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 em seu cluster for aplicativos lendo dados, você pode escalar um cluster Valkey ou Redis OSS (modo de cluster desativado) adicionando e excluindo réplicas de leitura. No entanto, há um máximo de 5 réplicas de leitura. Se a carga em seu cluster for pesada, você poderá se beneficiar dos endpoints de gravação adicionais de um cluster Valkey ou Redis OSS (modo de cluster ativado) 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.

## Minimizando o tempo de inatividade ElastiCache usando o Multi-AZ com Valkey e Redis OSS

Há vários casos em que, ElastiCache com o Valkey e o Redis, OSS pode ser necessário substituir um nó primário; isso inclui certos tipos de manutenção planejada e o evento improvável de uma falha no nó primário ou na 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, pois ele 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.

ElastiCache também propaga o nome Domain Name Service (DNS) 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 OSS clusters ElastiCache Valkey e Redis, as substituições planejadas dos nós são concluídas enquanto o cluster atende às solicitações de gravação recebidas.
- Para clusters desativados nos modos de OSS cluster Valkey e Redis com Multi-AZ habilitado e executados no mecanismo 5.0.6 ou posterior, as substituições planejadas dos nós são concluídas enquanto o cluster atende às solicitações de gravação recebidas.
- Para clusters desativados nos modos de OSS cluster Valkey e Redis com Multi-AZ habilitado que são executados no mecanismo 4.0.10 ou anterior, você pode notar uma breve interrupção de gravação associada às atualizações. 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.

Você pode ativar o Multi-AZ usando o ElastiCache Management Console AWS CLI, o ou o ElastiCache API

Habilitar o ElastiCache Multi-AZ em seu OSS cluster Valkey ou Redis (no grupo de replicação API e CLI,) melhora sua 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 compatível com OSS clusters Valkey e Redis 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 do Multi-AZ](#)

## Habilitar Multi-AZ

Você pode ativar o Multi-AZ ao criar ou modificar um cluster (API ou CLI grupo de replicação) usando o ElastiCache console ou o AWS CLI ElastiCache API

Você pode habilitar o Multi-AZ somente em clusters Valkey ou Redis OSS (modo de cluster desativado) que tenham 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 Valkey ou Redis OSS](#). Para obter informações sobre como adicionar uma réplica de leitura a um cluster com replicação, consulte [Adicionar uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#).

## Tópicos

- [Habilitação do multi-AZ \(console\)](#)
- [Habilitar o recurso multi-AZ \(AWS CLI\)](#)
- [Habilitar o recurso multi-AZ \(ElastiCache API\)](#)

## Habilitação do multi-AZ (console)

Você pode habilitar o Multi-AZ usando o ElastiCache console ao criar um novo cluster Valkey ou Redis ou ao modificar um OSS cluster existente com replicação.

O Multi-AZ é ativado por padrão nos clusters Valkey ou Redis OSS (modo de cluster ativado).

**⚠ Important**

ElastiCache ativará automaticamente o Multi-AZ somente se o cluster contiver pelo menos uma réplica em uma zona de disponibilidade diferente da primária em todos os fragmentos.

Ativando o Multi-AZ ao criar um cluster usando o console ElastiCache

Para obter mais informações sobre esse processo, consulte [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#). 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 Usando o ElastiCache AWS Management Console](#).

Habilitar o recurso multi-AZ (AWS CLI)

O exemplo de código a seguir usa o 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 JSON saída desse comando deve ser semelhante à 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 Referência de AWS CLI Comandos.

Habilitar o recurso multi-AZ (ElastiCache API)

O exemplo de código a seguir usa o ElastiCache API 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, consulte esses tópicos na ElastiCache APIReferência:

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

## Cenários de falha com respostas do multi-AZ

Antes da introdução do Multi-AZ, ElastiCache detectou e substituiu os nós com falha de um cluster recriando e reprovisionando o nó com falha. 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 ativado, monitora ElastiCache 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 ElastiCache Multi-AZ 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 seu aplicativo estiver gravando no endpoint primário, você não precisará alterar o endpoint para gravações ou leituras. ElastiCache propaga o DNS nome 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 — Você não precisa fazer nenhuma alteração em seu aplicativo, porque o DNS nome 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:

- [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
- [Encontrando os endpoints para grupos de OSS replicação Valkey ou Redis \(\)AWS CLI](#)
- [Encontrando endpoints para grupos de OSS replicação Valkey ou Redis \(\) ElastiCache API](#)

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 ElastiCache Multi-AZ 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 seu aplicativo estiver gravando no endpoint primário, não há necessidade de alterar o endpoint para gravações. ElastiCache propaga o DNS nome 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 DNS nome 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:

- [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
- [Encontrando os endpoints para grupos de OSS replicação Valkey ou Redis \(\)AWS CLI](#)
- [Encontrando endpoints para grupos de OSS replicação Valkey ou Redis \(\) ElastiCache API](#)

### 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 todo o cluster falha, o ElastiCache Multi-AZ 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:

- [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
- [Encontrando os endpoints para grupos de OSS replicação Valkey ou Redis \(\)AWS CLI](#)
- [Encontrando endpoints para grupos de OSS replicação Valkey ou Redis \(\) ElastiCache API](#)

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, você pode testá-lo usando o ElastiCache console AWS CLI, o. e. ElastiCache API

Ao testar, observe o seguinte:

- Você pode usar essa operação para testar o failover automático em até 15 fragmentos (chamados de grupos de nós no ElastiCache API e AWS CLI) em qualquer período contínuo de 24 horas.
- Se você chamar essa operação em fragmentos em clusters diferentes (chamados de grupos de replicação no API e CLI), poderá fazer as chamadas simultaneamente.
- Em alguns casos, você pode chamar essa operação várias vezes em fragmentos diferentes no mesmo grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). 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 foi concluída, verifique os eventos usando o ElastiCache console da Amazon AWS CLI, o ou ElastiCache API o. 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 as informações a seguir.

- [Visualizando ElastiCache eventos](#) no ElastiCache Guia do usuário
- [DescribeEvents](#) na ElastiCache API Referência
- [describe-events](#) na AWS CLI Referência de comandos.
- Isso foi API projetado para testar o comportamento do seu aplicativo em caso ElastiCache de falha. Ela não foi projetada para ser uma ferramenta operacional para iniciar um failover a fim de resolver um problema com o cluster. Além disso, em certas condições, como eventos operacionais de grande escala, AWS pode bloquear isso API.

## Tópicos

- [Testando o failover automático usando o AWS Management Console](#)
- [Testando o failover automático usando o AWS CLI](#)
- [Testando o failover automático usando o ElastiCache API](#)

## Testando o failover automático usando o AWS Management Console

Use o procedimento a seguir para testar o failover automático com o console.

Para testar o failover automático

1. Faça login no AWS Management Console e abra o ElastiCache console 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 você 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 [Usando o ElastiCache AWS Management Console](#).



5. Para Valkey ou Redis OSS (modo de cluster desativado), escolha o nome do cluster.

Para Valkey ou Redis OSS (modo de cluster ativado), faça o seguinte:

- a. Escolha o nome do cluster.

- b. Na página Shards, para o fragmento (chamado de grupo de nós no API eCLI) no qual você deseja testar o failover, escolha o nome do fragmento.
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).

## Testando o failover automático usando o AWS CLI

Você pode testar o failover automático em qualquer cluster habilitado para Multi-AZ usando a AWS CLI operação. `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 no máximo 15 grupos de nós em um período contínuo de 24 horas.

O exemplo a seguir usa o AWS CLI para testar o failover automático no grupo de nós `redis00-0003` no cluster Valkey ou Redis OSS (modo de cluster ativado). `redis00`

### Example 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 seu failover, use a AWS CLI `describe-events` operação.

Para obter mais informações, consulte as informações a seguir.

- [test-failover](#) na AWS CLI Referência de comandos.
- [describe-events](#) na AWS CLI Referência de comandos.

## Testando o failover automático usando o ElastiCache API

Você pode testar o failover automático em qualquer cluster habilitado com o Multi-AZ usando a ElastiCache API operação. `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 no máximo 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`.

### Example 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 seu failover, use a ElastiCache `DescribeEvents` API operação.

Para obter mais informações, consulte as informações a seguir.

- [TestFailover](#) na ElastiCache API Referência
- [DescribeEvents](#) na ElastiCache API Referência

## Limitações do Multi-AZ

Esteja ciente das seguintes limitações do Multi-AZ:

- O Multi-AZ é suportado no Valkey e no Redis OSS versão 2.8.6 e posterior.
- O Multi-AZ não é compatível com os tipos de nós T1.
- A OSS replicação do Valkey e do Redis 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, 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.

- Quando você promove manualmente réplicas de leitura para primárias em OSS clusters Valkey ou Redis com o modo de cluster desativado, você só pode fazer isso quando o failover automático e o Multi-AZ estão desativados. 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. Você pode fazer isso por meio do console desmarcando a caixa de seleção Failover automático do grupo de replicação. Você também pode fazer isso usando o AWS CLI definindo a `AutomaticFailoverEnabled` propriedade como `false` ao chamar a `ModifyReplicationGroup` operação.
  3. Promova a réplica de leitura para primário.
  4. Habilite novamente o Multi-AZ.
- ElastiCache (RedisOSS) Multi-AZ e arquivo somente para anexar (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. Em seguida, ElastiCache cria uma nova réplica no AZ-a (onde o primário com falha estava localizado) somente quando o AZ-a estiver de volta 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. ElastiCache Para obter detalhes sobre esse comportamento, consulte [Replicação](#) no site da Valkey.

 Important

Para Valkey 7.2.6 e posterior ou Redis OSS versão 2.8.22 e posterior, você não pode criar réplicas externas.

Para OSS versões do Redis anteriores à 2.8.22, recomendamos que você não conecte uma réplica externa a um ElastiCache cluster habilitado para Multi-AZ. Essa configuração não suportada pode criar problemas que impedem a execução adequada ElastiCache do failover e da recuperação. Para conectar uma réplica externa a um ElastiCache cluster, certifique-se de que o Multi-AZ não esteja habilitado antes de fazer a conexão.

## Como a sincronização e o backup são implementados

Todas as versões compatíveis do Valkey e do Redis OSS oferecem suporte ao backup e à 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

A OSS replicação do Redis, nas versões 2.8.22 e posteriores, escolha entre dois métodos. Para ter mais informações, consulte [OSSVersões do Redis antes de 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.

### OSSVersões do Redis antes de 2.8.22

O OSS backup e a sincronização do Redis nas versões anteriores à 2.8.22 são um processo de três etapas.

1. A bifurcação e, no processo em segundo plano, serializam os dados do cluster no disco. Isso cria um point-in-time instantâneo.
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 [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do 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 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 Valkey ou Redis disponível OSS (modo de cluster desativado) não associado a nenhum cluster que tenha réplicas para usar como nó primário. A outra é aplicada quando você precisa criar um nó primário com o cluster e as réplicas de leitura. Atualmente, um cluster Valkey ou Redis OSS (modo de cluster ativado) deve ser criado do zero.

### opção 1: [Criação de um grupo de replicação usando um cluster existente](#)

Use essa opção para aproveitar um cluster Valkey ou Redis de nó único existente OSS (modo de cluster desativado). 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

Você não pode criar um cluster Valkey ou Redis OSS (modo de cluster ativado) usando um cluster existente. Para criar um cluster Valkey ou Redis OSS (modo de cluster ativado) (API/CLI: grupo de replicação) usando o ElastiCache console, consulte [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

### opção 2: [Criando um grupo de OSS replicação Valkey ou Redis do zero](#)

Use essa opção se você ainda não tiver um cluster Valkey ou Redis disponível OSS (modo de cluster desativado) para usar como nó primário do cluster ou se quiser criar um cluster Valkey ou Redis OSS (modo de cluster ativado). Consulte [Criando um grupo de OSS replicação Valkey ou Redis do zero](#).

## Criação de um grupo de replicação usando um cluster existente

Um cluster disponível é um cluster Valkey ou OSS Redis de nó único existente. Atualmente, o Valkey ou o Redis OSS (modo de cluster ativado) não oferecem suporte à criação de um cluster com réplicas usando um cluster de nó único disponível. Se você quiser criar um cluster Valkey ou Redis OSS (modo de cluster ativado), consulte. [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

O procedimento a seguir só pode ser usado se você tiver um cluster de nó único Valkey ou Redis OSS (modo de cluster desativado). O nó desse cluster torna-se o primário no novo cluster. Se você não tiver um cluster Valkey ou Redis OSS (modo de cluster desativado) que possa ser usado como principal do novo cluster, consulte. [Criando um grupo de OSS replicação Valkey ou Redis do zero](#)

## Criação de um grupo de replicação usando um cluster existente (console)

Consulte o tópico [Usando o ElastiCache AWS Management Console](#).

## Criação de um grupo de replicação usando um cluster de OSS cache Valkey ou Redis disponível (AWS CLI)

Há duas etapas para criar um grupo de replicação com réplicas de leitura ao usar um cluster de OSS cache Valkey ou Redis disponível para o primário ao usar o 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 CLI comando, `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 Valkey ou Redis OSS (modo de cluster desativado) 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 nó do cluster Valkey ou Redis OSS (modo de cluster desativado) disponível que você deseja que seja o nó principal nesse grupo de replicação.

O comando a seguir cria o grupo de replicação `sample-repl-group` usando o cluster Valkey ou Redis disponível OSS (modo de cluster desativado) `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 de cache.

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 autônomo Valkey ou Redis OSS (modo de cluster desativado) ()  
ElastiCache API

Ao usar o 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 CLI comando, `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 Valkey ou Redis OSS (modo de cluster desativado) 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 Valkey ou Redis disponível OSS (modo de cluster desativado) que você deseja que seja o nó principal nesse cluster.

O comando a seguir cria o cluster com réplicas `sample-repl-group` usando o cluster Valkey ou Redis disponível OSS (modo de cluster desativado) `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 mais informações, consulte os tópicos da 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 de cache.

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 `myRep1Group`. 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 ElastiCache API tópico [CreateCacheCluster](#).

## Criando um grupo de OSS replicação Valkey ou Redis do zero

A seguir, você pode descobrir como criar um grupo de OSS replicação Valkey ou Redis sem usar um cluster Valkey ou Redis OSS existente como principal. Você pode criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado) do zero usando o ElastiCache console, o ou o. AWS CLI ElastiCache API

Antes de continuar, decida se deseja criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado). Para obter orientação nessa decisão, consulte [Replicação: modo de OSS cluster Valkey e Redis desativado versus ativado](#).

### Tópicos

- [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) do zero](#)
- [Criando um grupo de replicação no Valkey ou no Redis OSS \(modo de cluster ativado\) do zero](#)

## Criando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) do zero

Você pode criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) do zero usando o ElastiCache console, o AWS CLI, o AWS SDK ou o ElastiCache API. Um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) sempre tem um grupo de nós, um cluster primário e até cinco réplicas de leitura. Os grupos de replicação Valkey ou Redis OSS (modo de cluster desativado) não oferecem suporte ao particionamento de 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 Valkey ou Redis OSS (modo de cluster desativado) do zero, use uma das seguintes abordagens:

Criando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) do zero ( )AWS CLI

O procedimento a seguir cria um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) usando o AWS CLI

Ao criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) do zero, você cria o grupo de replicação e todos os seus nós com uma única chamada ao comando `AWS CLI create-replication-group`. 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 Valkey ou Redis OSS (modo de cluster desativado) 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.

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 geração anteriores.

Para obter mais informações sobre detalhes de desempenho para cada tipo de nó, consulte [Tipos de EC2 instância da Amazon](#).

**--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 [Hierarquização de dados em ElastiCache](#).

**--cache-parameter-group**

Especifique um grupo de parâmetros que corresponda à versão do seu mecanismo. Se você estiver executando o Redis OSS 3.2.4 ou posterior, especifique o grupo de `default.redis3.2` parâmetros ou um grupo de parâmetros derivado de `default.redis3.2` para criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado). Para obter mais informações, consulte [Parâmetros Valkey e 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 a OSS versão 3.2.6 ou 4.0.10 do Redis.
- O grupo de replicação deve ser criado em uma AmazonVPC.
- Você também deve incluir o parâmetro --cache-subnet-group.
- Você também deve incluir o parâmetro --auth-token com o valor da string especificado pelo cliente para seu AUTH token (senha) necessário para realizar operações nesse grupo de replicação.

A operação a seguir cria um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) 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 e parâmetros adicionais que você talvez queira usar, consulte o AWS CLI tópico [create-replication-group](#).

Criando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) do zero ()  
ElastiCache API

O procedimento a seguir cria um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) usando o. ElastiCache API

Ao criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) do zero, você cria o grupo de replicação e todos os seus nós com uma única chamada para a operação. ElastiCache API CreateReplicationGroup Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura de grupos de replicação Valkey ou Redis OSS (modo de cluster ativado) 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.

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 geração anteriores.

Para obter mais informações sobre detalhes de desempenho para cada tipo de nó, consulte [Tipos de EC2 instância da Amazon](#).

### --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 [Hierarquização de dados em ElastiCache](#).

### CacheParameterGroup

Especifique um grupo de parâmetros que corresponda à versão do seu mecanismo. Se você estiver executando o Redis OSS 3.2.4 ou posterior, especifique o grupo de `default.redis3.2` parâmetros ou um grupo de parâmetros derivado `default.redis3.2` para criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado). Para obter mais informações, consulte [Parâmetros Valkey e 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 a OSS versão 3.2.6 ou 4.0.10 do Redis.
- O grupo de replicação deve ser criado em uma AmazonVPC.
- Você também deve incluir o parâmetro `CacheSubnetGroup`.
- Você também deve incluir o parâmetro `AuthToken` com o valor da string especificado pelo cliente para seu AUTH token (senha) necessário para realizar operações nesse grupo de replicação.

A operação a seguir cria o grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) `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 e parâmetros adicionais que você talvez queira usar, consulte o ElastiCache API tópico [CreateReplicationGroup](#).

## Criando um grupo de replicação no Valkey ou no Redis OSS (modo de cluster ativado) do zero

Você pode criar um cluster Valkey ou Redis OSS (modo de cluster ativado) (API/CLI: grupo de replicação) usando o ElastiCache console AWS CLI, o ou o. ElastiCache API Um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) 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 nós ou fragmentos pode ser aumentado para um máximo de 500 por cluster se a versão do OSS mecanismo Valkey ou Redis 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 que as sub-redes no grupo de sub-redes têm um CIDR intervalo muito pequeno ou as sub-redes são compartilhadas e muito 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 no Redis OSS (modo de cluster ativado)

- [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)
- [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster ativado\) do zero \(\)AWS CLI](#)
- [Criando um grupo de replicação no Valkey ou no Redis OSS \(modo de cluster ativado\) do zero \(\)ElastiCache API](#)

### Criação de um cluster Valkey ou Redis OSS (modo de cluster ativado) (console)

Para criar um cluster Valkey ou Redis OSS (modo de cluster ativado), consulte. [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#) 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.

## Criando um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) do zero ()AWS CLI

O procedimento a seguir cria um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) usando o. AWS CLI

Ao criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) do zero, você cria o grupo de replicação e todos os seus nós com uma única chamada para o comando. AWS CLI `create-replication-group` Inclua os seguintes parâmetros.

`--replication-group-id`

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura de grupos de replicação Valkey ou Redis OSS (modo de cluster ativado) 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.

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 geração anteriores.

Para obter mais informações sobre os detalhes de desempenho de cada tipo de nó, consulte [Tipos de EC2 instância da Amazon](#).

`--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 [Hierarquização de dados em ElastiCache](#).

`--cache-parameter-group`

Especifique o grupo de `default.redis6.x.cluster.on` parâmetros ou um grupo de parâmetros derivado `default.redis6.x.cluster.on` para criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). Para obter mais informações, consulte [Alterações nos 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 a OSS versão 3.2.6 ou 4.0.10 do Redis.
- O grupo de replicação deve ser criado em uma AmazonVPC.

- Você também deve incluir o parâmetro `--cache-subnet-group`.
- Você também deve incluir o parâmetro `--auth-token` com o valor da string especificado pelo cliente para seu AUTH token (senha) necessário para realizar operações nesse grupo de replicação.

A operação a seguir cria o grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) `sample-repl-group` com três grupos/fragmentos de nós (`--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": {}
}

```

Ao criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) do zero, você pode configurar cada fragmento no cluster usando o `--node-group-configuration` parâmetro mostrado no 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, ElastiCache escolhe 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, ElastiCache escolhe 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, ElastiCache aloca as chaves igualmente entre os grupos de nós.

Exemplo: "0-4999"

A operação a seguir cria o grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) `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 e parâmetros adicionais que você talvez queira usar, consulte o AWS CLI tópico [create-replication-group](#).

Criando um grupo de replicação no Valkey ou no Redis OSS (modo de cluster ativado) do zero ()  
ElastiCache API

O procedimento a seguir cria um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) usando o. ElastiCache API

Ao criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) do zero, você cria o grupo de replicação e todos os seus nós com uma única chamada para a operação. ElastiCache API `CreateReplicationGroup` Inclua os seguintes parâmetros.

**ReplicationGroupId**

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura de grupos de replicação Valkey ou Redis OSS (modo de cluster ativado) 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, ElastiCache escolhe 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, ElastiCache escolhe 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, ElastiCache aloca as chaves igualmente entre os grupos de nós.

Exemplo: `"0-4999"`

## CacheNodeType

O tipo de nó para cada nó no grupo de replicação.

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 geração anteriores.

Para obter mais informações sobre os detalhes de desempenho de cada tipo de nó, consulte [Tipos de EC2 instância da Amazon](#).

### --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 [Hierarquização de dados em ElastiCache](#).

## CacheParameterGroup

Especifique o grupo de `default.redis6.x.cluster.on` parâmetros ou um grupo de parâmetros derivado `default.redis6.x.cluster.on` para criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). Para obter mais informações, consulte [Alterações nos 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 a OSS versão 3.2.6 ou 4.0.10 do Redis.
- O grupo de replicação deve ser criado em uma AmazonVPC.
- Você também deve incluir o parâmetro `CacheSubnetGroup`.

- Você também deve incluir o parâmetro AuthToken com o valor da string especificado pelo cliente para seu AUTH token (senha) necessário para realizar operações nesse 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 e parâmetros adicionais que você talvez queira usar, consulte o ElastiCache API tópico [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 ElastiCache console, o AWS CLI for ElastiCache ou ElastiCache API o. O processo do console é diferente para Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado).

Visualizar os detalhes de um grupo de replicação

- [Visualizando um Valkey ou Redis OSS \(modo de cluster desativado\) com réplicas](#)
  - [Visualizando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
  - [Visualizando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) \(AWS CLI\)](#)
  - [Visualizando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) \(ElastiCache API\)](#)

- [Visualizando um grupo de replicação: Valkey ou Redis OSS \(modo de cluster ativado\)](#)
  - [Visualizando um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)
  - [Visualizando um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(\)AWS CLI](#)
  - [Visualizando um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(\) ElastiCache API](#)
- [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#)
- [Visualização dos detalhes de um grupo de replicação \(ElastiCache API\)](#)

Visualizando um Valkey ou Redis OSS (modo de cluster desativado) com réplicas

Você pode ver os detalhes de um cluster Valkey ou Redis OSS (modo de cluster desativado) com réplicas (API/CLI: grupo de replicação) usando o ElastiCache console, o AWS CLI for ElastiCache ou o ElastiCache API

Visualizando detalhes de um cluster Valkey ou Redis OSS (modo de cluster desativado)

- [Visualizando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
- [Visualizando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) \(\)AWS CLI](#)
- [Visualizando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) \(\) ElastiCache API](#)

Visualizando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) (console)

Para ver os detalhes de um cluster Valkey ou Redis OSS (modo de cluster desativado) com réplicas usando o ElastiCache console, consulte o tópico. [Visualizando detalhes do Valkey ou do Redis OSS \(modo de cluster desativado\) \(console\)](#)

Visualizando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) ()AWS CLI

Para obter um AWS CLI exemplo que exibe os detalhes de um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado), consulte. [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#)

## Visualizando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) () ElastiCache API

Para obter um ElastiCache API exemplo que exibe os detalhes de um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado), consulte. [Visualização dos detalhes de um grupo de replicação \(ElastiCache API\)](#)

## Visualizando um grupo de replicação: Valkey ou Redis OSS (modo de cluster ativado)

### Visualizando um cluster Valkey ou Redis OSS (modo de cluster ativado) (console)

Para ver os detalhes de um cluster Valkey ou Redis OSS (modo de cluster ativado) usando o ElastiCache console, consulte. [Visualizando detalhes de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

### Visualizando um cluster Valkey ou Redis OSS (modo de cluster ativado) ()AWS CLI

Para obter um ElastiCache CLI exemplo que exibe os detalhes de um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado), consulte. [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#)

### Visualizando um cluster Valkey ou Redis OSS (modo de cluster ativado) () ElastiCache API

Para obter um ElastiCache API exemplo que exibe os detalhes de um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado), consulte. [Visualização dos detalhes de um grupo de replicação \(ElastiCache API\)](#)

### Visualização dos detalhes de um grupo de replicação (AWS CLI)

Você pode ver os detalhes de um grupo de replicação usando o AWS CLI `describe-replication-groups` comando. 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
```

A saída dessa operação deve ser semelhante a esta (JSONformato).

```
{
  "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 o ElastiCache tópico AWS CLI for [describe-replication-groups](#).

## Visualização dos detalhes de um grupo de replicação (ElastiCache API)

Você pode ver os detalhes de uma replicação usando a AWS CLI `DescribeReplicationGroups` operação. 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 ElastiCache API de referência [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 DNS endpoint e o número da porta desse nó. Dependendo se você estiver executando um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) ou Valkey ou Redis OSS (modo de cluster ativado), você estará interessado em endpoints diferentes.

### Valkey ou Redis OSS (modo de cluster desativado)

Os clusters Valkey ou Redis OSS (modo de cluster desativado) com réplicas têm três tipos de endpoints: o endpoint primário, o endpoint do leitor e os endpoints do nó. O endpoint primário é um DNS nome 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 cluster ElastiCache (RedisOSS) 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 DNS registro que será resolvido para um endereço IP de um dos nós de réplica de forma redonda.

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 de cluster ativado)

Clusters Valkey ou Redis OSS (modo de cluster ativado) com réplicas, porque eles têm vários fragmentos (API/CLI: grupos de nós), o que significa que eles também têm vários nós primários, têm uma estrutura de endpoint diferente dos clusters Valkey ou Redis OSS (modo de cluster

desativado). O Valkey ou Redis OSS (modo de cluster ativado) tem um endpoint de configuração que “conhece” todos os endpoints primários e de nós no cluster. Seu aplicativo conecta-se ao endpoint de configuração. Sempre que seu aplicativo grava ou lê a partir do endpoint de configuração do cluster, o Valkey e o RedisOSS, nos bastidores, determinam a qual fragmento a chave pertence e qual endpoint desse fragmento usar. O processo é bastante transparente para o seu aplicativo.

Você pode encontrar os endpoints de um cluster usando o ElastiCache console AWS CLI, o ou o. ElastiCache API

### Localizar endpoints de grupo de replicação

Para localizar os endpoints do seu grupo de replicação, consulte um dos seguintes tópicos:

- [Encontrando os endpoints de um cluster Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
- [Encontrando endpoints para um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)
- [Encontrando os endpoints para grupos de OSS replicação Valkey ou Redis \(\)AWS CLI](#)
- [Encontrando endpoints para grupos de OSS replicação Valkey ou Redis \(\) ElastiCache API](#)

## 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 API operação `ModifyReplicationGroup` (CLI): `modify-replication-group`. Você pode modificar o número de fragmentos (grupos de nós) em um cluster Valkey ou Redis OSS (modo de cluster ativado) com a API operação [ModifyReplicationGroupShardConfiguration](#) (CLI): `modify-replication-group-shard-configuration`. Para obter mais informações, consulte [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#).

Outras modificações em um cluster Valkey ou Redis OSS (modo de cluster ativado) exigem que você crie um cluster com o novo cluster incorporando as alterações.

- Você pode atualizar clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster desativado) e Valkey ou Redis OSS (modo de cluster ativado) 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 OSS cluster existente ElastiCache com Valkey ou Redis que usa o modo de cluster desativado para usar o modo de cluster ativado, usando o console [ModifyReplicationGroup](#) API ou o `modify-replication-group` CLI comando, 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 o. ElastiCache 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 atual e, em seguida, use esse backup para implantar um novo grupo de replicação Valkey ou Redis OSS (modo de cluster ativado).

### Tópicos

- [Usando o AWS Management Console](#)

- [Usando o AWS CLI](#)
- [Usando o ElastiCache API](#)

## Usando o AWS Management Console

Para modificar um cluster Valkey ou Redis OSS (modo de cluster desativado), consulte. [Modificando um cluster 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.

Ative o Multi-AZ em um grupo de replicação existente do Valkey ou do RedisOSS:

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 OSS clientes Valkey ou Redis se conectem usando o modo de cluster ativado e o modo de cluster desativado. Depois de migrar todos os OSS clientes Valkey ou Redis para usar o modo de cluster ativado, você pode concluir a configuração do modo de cluster e definir o modo de cluster como ativado.

Para Linux, macOS ou Unix:

Defina o modo de cluster como compatível.

```
aws elasticache modify-replication-group \  
  --multi-az-enabled
```



```
--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 usuário ElastiCache (RedisOSS).

## Usando o ElastiCache API

A ElastiCache API operação a seguir ativa 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 ElastiCache API `ModifyReplicationGroup` operação, 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 no API/CLI), poderá excluí-lo. Quando você exclui um grupo de replicação, ElastiCache exclui todos os nós desse grupo.

Depois de ter iniciado essa operação, ela não poderá ser interrompida ou cancelada.

### Warning

- Quando você exclui um cluster ElastiCache (RedisOSS), seus instantâneos manuais são retidos. Você também poderá criar um snapshot final antes da exclusão do cluster. Os snapshots de cache automáticos não são retidos.
- `CreateSnapshot` é necessária permissão para criar um instantâneo final. Sem essa permissão, a API chamada falhará com uma `Access Denied` exceção.

### 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
```

### Excluindo um grupo de replicação () ElastiCache API

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 em seu grupo de OSS replicação Valkey ou Redis usando o AWS Management Console, o ou o. AWS CLI ElastiCache API Se seu grupo de replicação for um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado), você poderá escolher quais fragmentos (grupos de nós) aumentar ou diminuir o número de réplicas.

Para alterar dinamicamente o número de réplicas em seu grupo de replicação, escolha a operação na tabela a seguir que se adapte à sua situação.

| Para fazer isso    | Para Valkey ou Redis OSS (modo de cluster ativado)               | Para Valkey ou Redis OSS (modo de cluster desativado)                                                                                                                          |
|--------------------|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adicionar réplicas | <a href="#">Aumento do número de réplicas em um fragmento</a>    | <a href="#">Aumento do número de réplicas em um fragmento</a><br><br><a href="#">Adicionar uma réplica de leitura para Valkey ou Redis OSS (modo de cluster desativado)</a>    |
| Excluir réplicas   | <a href="#">Diminuição do número de réplicas em um fragmento</a> | <a href="#">Diminuição do número de réplicas em um fragmento</a><br><br><a href="#">Excluindo uma réplica de leitura para Valkey ou Redis OSS (modo de cluster desativado)</a> |

## Aumento do número de réplicas em um fragmento

Você pode aumentar o número de réplicas em um fragmento Valkey ou Redis OSS (modo de cluster ativado) ou grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) até um máximo de cinco. Você pode fazer isso usando o AWS Management Console AWS CLI, o ou ElastiCache API o.

### Tópicos

- [Usando o AWS Management Console](#)
- [Usando o AWS CLI](#)
- [Usando o ElastiCache API](#)

### Usando o AWS Management Console

O procedimento a seguir usa o console para aumentar o número de réplicas em um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado).

Para aumentar o número de réplicas em fragmentos

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis eOSS, em seguida, escolha 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.
  - Para Zonas de disponibilidade, escolha Sem preferência para ElastiCache escolher uma zona de disponibilidade para cada nova réplica ou Especificar zonas de disponibilidade para 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.

## Usando o AWS CLI

Para aumentar o número de réplicas em um OSS fragmento do Valkey ou do Redis, use o `increase-replica-count` comando 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`: optional. Especifica o número de nós de réplica que você deseja concluir, até um máximo de cinco. Use esse parâmetro para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado) onde há somente um grupo de nós ou grupo Valkey ou Redis OSS (modo de cluster ativado), 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`: optional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use esse parâmetro para grupos Valkey ou Redis OSS (modo de cluster ativado) nos quais você deseja configurar cada grupo de nós de forma independente.

`--replica-configuration` tem três membros opcionais:

- `NodeGroupId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado), o ID do fragmento é sempre `0001`. Para encontrar a ID de um grupo de nós OSS (fragmento) do Valkey ou do Redis (modo de cluster ativado), 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. Se esse membro de --replica-configuration for omitido, ElastiCache (RedisOSS) 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 se for um grupo Valkey ou Redis OSS (modo de cluster desativado) com um único grupo de nós ou um grupo Valkey ou Redis OSS (modo de cluster ativado) 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, esse é um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). 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 o CLI, consulte [increase-replica-count](#) Referência da Linha de ElastiCache Comando da Amazon.

Usando o ElastiCache API

Para aumentar o número de réplicas em um OSS fragmento do Valkey ou do Redis, use a `IncreaseReplicaCount` ação 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`: optional. Especifica o número de nós de réplica que você deseja concluir, até um máximo de cinco. Use esse parâmetro para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado) em que há apenas um grupo de nós, ou grupos Valkey ou Redis OSS (modo de cluster ativado) em que 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`: optional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use esse parâmetro para grupos Valkey ou Redis OSS (modo de cluster ativado) nos quais você deseja configurar cada grupo de nós de forma independente.

`ReplicaConfiguration` tem três membros opcionais:

- `NodeId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado), a ID do grupo de nós (fragmento) é sempre `0001`. Para encontrar a ID de um grupo de nós OSS (fragmento) do Valkey ou do Redis (modo de cluster ativado), 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. Se esse membro de `ReplicaConfiguration` for omitido, ElastiCache (RedisOSS) 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 se for um grupo Valkey ou Redis OSS (modo de cluster desativado) com um único grupo de nós ou um grupo Valkey ou Redis OSS (modo de cluster ativado) 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, esse é um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). 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 o API, consulte [IncreaseReplicaCount](#) na Amazon ElastiCache API Reference.

## Diminuição do número de réplicas em um fragmento

Você pode diminuir o número de réplicas em um fragmento para Valkey ou Redis OSS (modo de cluster ativado) ou em um grupo de replicação para Valkey ou OSS Redis (modo de cluster desativado):

- Para Valkey ou Redis OSS (modo de cluster desativado), você pode diminuir o número de réplicas para uma se o Multi-AZ estiver ativado e para zero se não estiver ativado.
- Para Valkey ou Redis OSS (modo de cluster ativado), 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 AWS Management Console, o AWS CLI ou o ElastiCache API para diminuir o número de réplicas em um grupo de nós (fragmento) ou grupo de replicação.

### Tópicos

- [Usando o AWS Management Console](#)
- [Usando o AWS CLI](#)
- [Usando o ElastiCache API](#)

### Usando o AWS Management Console

O procedimento a seguir usa o console para diminuir o número de réplicas em um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado).

Para diminuir o número de réplicas em um fragmento Valkey ou Redis OSS

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Valkey ou Redis e, em seguida OSS, escolha o nome do grupo de replicação do qual você deseja excluir as 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

- Se você não especificar os nós de réplica a serem excluídos, o ElastiCache (RedisOSS) selecionará automaticamente os nós de réplica para exclusão. Ao fazer isso, o ElastiCache (RedisOSS) tenta manter a arquitetura Multi-AZ para seu grupo de replicação, seguida pela retenção de réplicas com atraso mínimo de replicação com a primária.
- 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.

## Usando o AWS CLI

Para diminuir o número de réplicas em um OSS fragmento do Valkey ou do Redis, use o `decrease-replica-count` comando 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`: optional. 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 chama falha.
- `--replicas-to-remove`: optional. Contém uma lista de nós IDs especificando os nós de réplica a serem removidos.
- `--replica-configuration`: optional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use esse parâmetro para grupos Valkey ou Redis OSS (modo de cluster ativado) nos quais você deseja configurar cada grupo de nós de forma independente.

`--replica-configuration` tem três membros opcionais:

- `NodeGroupId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado), o ID do fragmento é sempre `0001`. Para encontrar a ID de um grupo de nós OSS (fragmento) do Valkey ou do Redis (modo de cluster ativado), 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. Se esse membro de `--replica-configuration` for omitido, ElastiCache (RedisOSS) 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 se for um grupo Valkey ou Redis OSS (modo de cluster desativado) com um único grupo de nós ou um grupo Valkey ou Redis OSS (modo de cluster ativado) 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, esse é um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). 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 \
    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 diminuir o número de réplicas usando o CLI, consulte a Referência da Linha [decrease-replica-count](#) de ElastiCache Comando da Amazon.

### Usando o ElastiCache API

Para diminuir o número de réplicas em um OSS fragmento do Valkey ou do Redis, use a `DecreaseReplicaCount` ação 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`: optional. 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 chama falha.
- `ReplicasToRemove`: optional. Contém uma lista de nós IDs especificando os nós de réplica a serem removidos.
- `ReplicaConfiguration`: optional. 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 esse parâmetro para grupos Valkey ou Redis OSS (modo de cluster ativado) nos quais você deseja configurar cada grupo de nós de forma independente.

ReplicaConfiguration tem três membros opcionais:

- **NodeId:** o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado), o ID do grupo de nós é sempre `0001`. Para encontrar a ID de um grupo de nós OSS (fragmento) do Valkey ou do Redis (modo de cluster ativado), 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. Se esse membro de `ReplicaConfiguration` for omitido, ElastiCache (RedisOSS) 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 se for um grupo Valkey ou Redis OSS (modo de cluster desativado) com um único grupo de nós ou um grupo Valkey ou Redis OSS (modo de cluster ativado) 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, esse é um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado). 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 diminuir o número de réplicas usando o API, consulte [DecreaseReplicaCount](#) na Referência da Amazon ElastiCache API.

Adicionar uma réplica de leitura para Valkey ou Redis OSS (modo de cluster desativado)

As informações no tópico a seguir se aplicam somente aos grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).

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 Valkey ou Redis OSS (modo de cluster desativado).

Um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) 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 Valkey ou Redis OSS (modo de cluster ativado), consulte o seguinte:

- [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)
- [Aumento do número de réplicas em um fragmento](#)

Você pode adicionar uma réplica de leitura a um cluster Valkey ou Redis OSS (modo de cluster desativado) usando o ElastiCache console AWS CLI, o ou o. ElastiCache API

## Tópicos relacionados da

- [Adicionando nós a um ElastiCache cluster](#)
- [Adição de uma réplica de leitura a um grupo de replicação \(AWS CLI\)](#)
- [Adicionar uma réplica de leitura a um grupo de replicação usando o 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 Valkey ou Redis OSS (modo de cluster desativado), use o AWS CLI `create-cache-cluster` comando, 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 o CLI, consulte [create-cache-cluster](#) na Amazon ElastiCache Command Line Reference.

### Adicionar uma réplica de leitura a um grupo de replicação usando o API

Para adicionar uma réplica de leitura a um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado), use a ElastiCache `CreateCacheCluster` operação, com o parâmetro `ReplicationGroupId` para especificar em 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 o API, consulte [CreateCacheCluster](#) na Amazon ElastiCache API Reference.

Excluindo uma réplica de leitura para Valkey ou Redis OSS (modo de cluster desativado)

As informações no tópico a seguir se aplicam somente aos grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).

Conforme o tráfego de leitura em seu grupo de OSS replicação Valkey ou Redis muda, 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 [Promovendo uma réplica de leitura para primária, para grupos de replicação Valkey ou Redis OSS \(modo de cluster desativado\)](#).
  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 Valkey ou Redis OSS (modo de cluster desativado) usando o ElastiCache console, o AWS CLI for ElastiCache ou o ElastiCache API

Para obter instruções sobre como excluir um cluster de um grupo de OSS replicação Valkey ou Redis, consulte o seguinte:

- [Usando o AWS Management Console](#)
- [Usando o AWS CLI para excluir um ElastiCache cluster](#)
- [Usando o ElastiCache API](#)
- [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)
- [Diminuição do número de réplicas em um fragmento](#)

## Promovendo uma réplica de leitura para primária, para grupos de replicação Valkey ou Redis OSS (modo de cluster desativado)

As informações no tópico a seguir se aplicam somente aos grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).

Você pode promover uma réplica de leitura Valkey ou Redis OSS (modo de cluster desativado) para primária usando o AWS Management Console AWS CLI, o ou o. ElastiCache API 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 Valkey ou Redis OSS (modo de cluster desativado) para primária 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 Redis OSS 2.6.13 ou anterior.

Usando o AWS Management Console

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 AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Se a réplica que você deseja promover for membro de um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) em que o Multi-AZ esteja habilitado, modifique o grupo de replicação para desativar o Multi-AZ antes de continuar. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
3. Escolha Valkey ou Redis eOSS, 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](#).

## Usando o 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 AWS CLI comando a seguir modifica o grupo de replicação `sample-repl-group`, tornando a réplica de leitura `my-replica-1` a principal 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 a modificação de um grupo de replicação, consulte [modify-replication-group](#) na Amazon ElastiCache Command Line Reference.

## Usando o ElastiCache API

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 ElastiCache API ação a seguir modifica o grupo de replicação `myRep1Group`, tornando a réplica de leitura `myReplica-1` a principal no grupo de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ReplicationGroupId=myRep1Group  
&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 a modificação de um grupo de replicação, consulte [ModifyReplicationGroup](#) na Amazon ElastiCache API Reference.

## Gerenciando a manutenção ElastiCache do cluster

Cada cluster tem uma janela de manutenção semanal durante a qual todas as alterações do sistema são aplicadas. Com o Valkey e o RedisOSS, 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, ElastiCache atribua uma janela de manutenção

de 60 minutos dentro da janela de manutenção da 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)    | 13:00 — 21:00 UTC              |
| ap-northeast-2   | Região Ásia-Pacífico (Seul)      | 12:00 — 20:00 UTC              |
| ap-northeast-3   | Região Ásia-Pacífico (Osaka)     | 12:00 — 20:00 UTC              |
| ap-southeast-3   | Região Ásia-Pacífico (Jacarta)   | 14:00 — 22:00 UTC              |
| ap-south-1       | Região Ásia-Pacífico (Mumbai)    | 17:30 — 1:30 UTC               |
| ap-southeast-1   | Região Ásia-Pacífico (Singapura) | 14:00 — 22:00 UTC              |
| cn-north-1       | Região China (Pequim)            | 14:00 — 22:00 UTC              |
| cn-northwest-1   | Região China (Ningxia)           | 14:00 — 22:00 UTC              |
| ap-east-1        | Região Ásia-Pacífico (Hong Kong) | 13:00 — 21:00 UTC              |
| ap-southeast-2   | Região Ásia-Pacífico (Sydney)    | 12:00 — 20:00 UTC              |
| eu-west-3        | Região Europa (Paris)            | 23:59 — 07:29 UTC              |
| af-south-1       | Região África (Cidade do Cabo)   | 13:00 — 21:00 UTC              |
| eu-central-1     | Região Europa (Frankfurt)        | 23:00 — 07:00 UTC              |
| eu-west-1        | Região Europa (Irlanda)          | 22:00 — 06:00 UTC              |
| eu-west-2        | Região Europa (Londres)          | 23:00 — 07:00 UTC              |
| me-south-1       | Região Oriente Médio (Bahrein)   | 13:00 — 21:00 UTC              |

| Código da região | Nome da região                             | Janela de manutenção da região |
|------------------|--------------------------------------------|--------------------------------|
| me-central-1     | Região do Oriente Médio (UAE)              | 13:00 — 21:00 UTC              |
| eu-south-1       | Região Europa (Milão)                      | 21:00 — 05:00 UTC              |
| sa-east-1        | Região América do Sul (São Paulo)          | 01:00 — 09:00 UTC              |
| us-east-1        | Região Leste dos EUA (N. da Virgínia)      | 03:00 — 11:00 UTC              |
| us-east-2        | Região Leste dos EUA (Ohio)                | 04:00 — 12:00 UTC              |
| us-gov-west-1    | AWS GovCloud (US) região                   | 06:00 — 14:00 UTC              |
| us-west-1        | Região Oeste dos EUA (Norte da Califórnia) | 06:00 — 14:00 UTC              |
| us-west-2        | Região Oeste dos EUA (Oregon)              | 06:00 — 14:00 UTC              |

## Alterando a janela de manutenção do seu 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 a AWS Management Console caixa Aplicar agora. Do contrário, essas modificações serão aplicadas durante o próximo período de manutenção programada. Para usar oAPI, consulte [modify-replication-group](#) ou [modify-cache-cluster](#).

## Mais informações

Para obter informações sobre sua janela de manutenção e substituição de nó, consulte:

- [ElastiCache Manutenção](#) — FAQ sobre manutenção e substituição de nós
- [Substituindo nós \(Memcached\)](#)—Gerenciando a substituição de nós para o Memcached
- [Modificando um cluster ElastiCache](#) : Alteração da janela de manutenção de um cluster
- [Substituindo nós \(Valkey e RedisOSS\)](#): Gerenciamento de substituição de nó
- [Modificação de um grupo de replicação](#): Alteração da janela de manutenção de um grupo de replicação

## Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros

A 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 OSS parâmetros do Valkey e do Redis, consulte. [Parâmetros Valkey e Redis OSS](#)

Como seria de se esperar, alguns valores de parâmetros, como `maxmemory`, são determinados pelo mecanismo e tipo de nó. Para obter uma tabela desses valores de parâmetros do Memcached por tipo de nó, consulte. [Parâmetros específicos do tipo de nó do Memcached](#) Para obter uma tabela desses valores de OSS parâmetros Valkey e Redis por tipo de nó, consulte. [Parâmetros específicos do tipo de OSS nó Redis](#)

### Note

Para obter uma lista de parâmetros específicos do Memcached, consulte [Parâmetros específicos de Memcached](#).

### Tópicos

- [Gerenciamento de parâmetros em ElastiCache](#)
- [Camadas de grupos de parâmetros de cache em ElastiCache](#)
- [Criação de um grupo de ElastiCache parâmetros](#)
- [Listando grupos de ElastiCache parâmetros por nome](#)

- [Listando os valores de um grupo de ElastiCache parâmetros](#)
- [Modificando um grupo de ElastiCache parâmetros](#)
- [Excluindo um grupo de ElastiCache parâmetros](#)
- [Parâmetros específicos do motor](#)

## Gerenciamento de parâmetros em ElastiCache

ElastiCache os parâmetros 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 de grupos de parâmetros e o cluster ao qual você 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, a família de grupos de parâmetros e o cluster ao qual você está atribuindo devem ser compatíveis. Por exemplo, se seu cluster estiver executando a OSS versão 3.2.10 do Redis, 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 imediatamente ao cluster. 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, quando você altera os parâmetros de um cluster, a alteração é aplicada ao cluster imediatamente ou, com as exceções mencionadas a seguir, após a reinicialização dos nós do cluster. 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 Valkey e Redis OSS](#).

Para obter mais informações sobre a reinicialização dos nós Valkey ou RedisOSS, consulte.

## [Reinicializando nós](#)

### Alterações nos parâmetros Valkey ou Redis OSS (modo de cluster ativado)

Se você fizer alterações nos seguintes parâmetros em um cluster Valkey ou Redis OSS (modo de cluster ativado), 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 aos datastores OSS globais Valkey e Redis. Os armazenamentos de dados globais são uma coleção de um ou mais clusters que AWS abrangem regiões. 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](#).

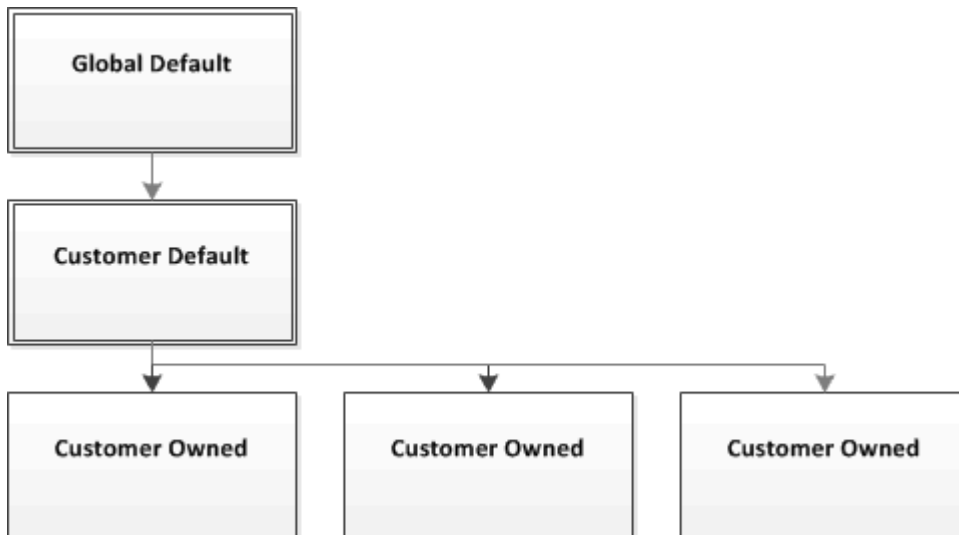
Você pode verificar se um grupo de parâmetros faz parte de um datastore global procurando nestes locais:

- No ElastiCache console, na página Parameter Groups, o atributo global sim/não
- A `IsGlobal` propriedade sim/não da operação [CacheParameterGroupAPI](#)



## Camadas de grupos de parâmetros de cache em ElastiCache

ElastiCache A Amazon tem três níveis de grupos de parâmetros de cache, conforme mostrado a seguir.



### Níveis de grupos de ElastiCache parâmetros da Amazon

#### Padrão global

O grupo de parâmetros raiz de nível superior para todos os ElastiCache clientes da Amazon na região.

O parameter group de cache padrão global:

- Está reservado 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 de 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 os valores do Memcached, consulte [Parâmetros específicos do Memcached](#). Para obter mais informações sobre os OSS valores do Valkey e do Redis, consulte [Parâmetros Valkey e Redis OSS](#)

## Criação de um grupo de ElastiCache parâmetros

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 ElastiCache console AWS CLI, o ou ElastiCache API o.

### Criação de um grupo de ElastiCache parâmetros (Console)

O procedimento a seguir mostra como criar um grupo de parâmetros usando o ElastiCache console.

Para criar um grupo de parâmetros usando o ElastiCache console

1. Faça login no AWS Management Console e abra o ElastiCache console 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 grupos de parâmetros, como memcached1.4 ou redis3.2, define os parâmetros reais em seu grupo de parâmetros e 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 ASCII carta.
  - Só pode conter ASCII letras, dígitos e hífen.
  - Deve ter entre 1 e 255 caracteres.
  - Não podem conter dois hífen 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 [Modificando um grupo de ElastiCache parâmetros](#).

### Criando um grupo de ElastiCache parâmetros (AWS CLI)

Para criar um grupo de parâmetros usando o 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 ASCII carta.
  - Só pode conter ASCII letras, dígitos e hífen.
  - Deve ter entre 1 e 255 caracteres.
  - Não podem conter dois hífen 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 grupo de parâmetros 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 grupo de parâmetros 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"
```

```
--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 [Modificando um grupo de ElastiCache parâmetros](#).

Para obter mais informações, consulte [create-cache-parameter-group](#).

Criando um grupo de ElastiCache parâmetros (ElastiCache API)

Para criar um grupo de parâmetros usando o ElastiCache API, use a `CreateCacheParameterGroup` ação 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 ASCII carta.
- Só pode conter ASCII letras, dígitos e hífens.
- 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 grupo de parâmetros 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 grupo de parâmetros 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 [Modificando um grupo de ElastiCache parâmetros](#).

Para obter mais informações, consulte [CreateCacheParameterGroup](#).

## Listando grupos de ElastiCache parâmetros por nome

Você pode listar os grupos de parâmetros usando o ElastiCache console, o AWS CLI, o ou ElastiCache API.

### Listagem de grupos de parâmetros por nome (console)

O procedimento a seguir mostra como exibir uma lista dos grupos de parâmetros usando o ElastiCache console.

Para listar grupos de parâmetros usando o ElastiCache console

1. Faça login no AWS Management Console e abra o ElastiCache console 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.

### Listando grupos de ElastiCache parâmetros por nome (AWS CLI)

Para gerar uma lista de grupos de parâmetros usando o 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 grupo de parâmetros `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 grupo de parâmetros 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 executados no OSS mecanismo Redis da versão 5.0.6 em diante. 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 semelhante a esta, listando o nome, a família `isGlobal` e a 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 JSON saída desse comando será mais ou menos assim, listando o nome, a família, a descrição e, no caso do `redis5.6`, se o grupo de parâmetros faz parte de um armazenamento de dados 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](#).

## Listando grupos de ElastiCache parâmetros por nome (ElastiCache API)

Para gerar uma lista de grupos de parâmetros usando o ElastiCache API, use a `DescribeCacheParameterGroups` ação. 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 grupo de parâmetros `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 mais ou menos assim, listando o nome, a família, a descrição e, no caso de redis5.6, se o grupo de parâmetros pertencer a um armazenamento de dados global (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 grupo de parâmetros 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 dessa ação será mais ou menos assim, listando o nome, a família, a descrição e se o grupo de parâmetros faz parte de um armazenamento de dados 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](#).

## Listando os valores de um grupo de ElastiCache parâmetros

Você pode listar os parâmetros e seus valores para um grupo de parâmetros usando o ElastiCache console, o AWS CLI, ou o ElastiCache API.

### Listando os valores de um grupo de ElastiCache parâmetros (Console)

O procedimento a seguir mostra como listar os parâmetros e seus valores para um grupo de parâmetros usando o ElastiCache console.

Para listar os parâmetros de um grupo de parâmetros e seus valores usando o ElastiCache console

1. Faça login no AWS Management Console e abra o ElastiCache console 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 o 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 myMem de parâmetros 14.

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 grupo de parâmetros 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 (ElastiCache API)

Para listar os parâmetros de um grupo de parâmetros e seus valores usando o ElastiCache API, use a DescribeCacheParameters ação.

## Example

O código de exemplo a seguir lista todos os parâmetros do Memcached para o grupo myMem de parâmetros 14.

```
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>
      </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 do grupo de parâmetros 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>
      </CacheClusterClassSpecificParameters>
    </DescribeCacheParametersResult>
    <ResponseMetadata>
      <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
    </ResponseMetadata>
  </DescribeCacheParametersResponse>

```

...output omitted...

```
</ResponseMetadata>  
</DescribeCacheParametersResponse>
```

Para obter mais informações, consulte [DescribeCacheParameters](#).

## Modificando um grupo de ElastiCache parâmetros

### 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 Valkey e Redis OSS](#) e [Parâmetros específicos do Memcached](#).

### Modificação de um grupo de parâmetros (console)

O procedimento a seguir mostra como alterar o valor do `cluster-enabled` parâmetro usando o ElastiCache console. Você usaria o mesmo procedimento para alterar o valor de qualquer parâmetro.

Para alterar o valor de um parâmetro usando o ElastiCache console


1. Faça login no AWS Management Console e abra o ElastiCache console 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 Save Changes (Salvar alterações).

7. Para Memcached, para encontrar 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 RedisOSS, para encontrar o nome do parâmetro que você alterou, consulte [Parâmetros Valkey e Redis OSS](#). Se você tiver um cluster Valkey ou Redis OSS (modo de cluster desativado) e fizer alterações nos seguintes parâmetros, deverá reinicializar os nós no cluster:
  - activerehashing
  - bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

 Alterações nos parâmetros Valkey ou Redis OSS (modo de cluster ativado)

Se você fizer alterações nos seguintes parâmetros em um cluster Valkey ou Redis OSS (modo de cluster ativado), siga as etapas a seguir.

- activerehashing
  - bancos de dados
1. Com o Redis, você pode criar 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 (AWS CLI)

Para alterar o valor de um parâmetro usando o AWS CLI, use o comando `modify-cache-parameter-group`.

## Example

Com o Memcached, para encontrar 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 Valkey e RedisOSS, para encontrar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros Valkey e Redis OSS](#)

O código de exemplo a seguir define o valor de dois parâmetros `reserved-memory-percent` e `cluster-enabled` no grupo de parâmetros `myredis32-on-30`. Definimos `reserved-memory-percent30` (30 por cento) e habilitado para cluster para que o grupo de parâmetros `yes` possa ser usado com clusters Valkey ou Redis OSS (habilitado no modo de cluster) (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 Valkey e Redis OSS](#).

Se você tiver um cluster Valkey ou Redis OSS (modo de cluster desativado) e fizer alterações nos seguintes parâmetros, deverá reinicializar os nós no cluster:

- activerehashing
- bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

#### Alterações nos parâmetros Valkey ou Redis OSS (modo de cluster ativado)

Se você fizer alterações nos seguintes parâmetros em um cluster Valkey ou Redis OSS (modo de cluster ativado), 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 (ElastiCache API)

Para alterar os valores de parâmetros de um grupo de parâmetros usando o ElastiCache API, use a `ModifyCacheParameterGroup` ação.

### Example

Com o Memcached, para encontrar 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 Valkey e RedisOSS, para encontrar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros Valkey e Redis OSS](#)



O código de exemplo a seguir define o valor de dois parâmetros `reserved-memory-percent` e `cluster-enabled` é ativado por cluster no grupo de parâmetros `myredis32-on-30`. Definimos `reserved-memory-percent` como `reserved-memory-percent30` (30 por cento) e habilitado para cluster para que o grupo de parâmetros `yes` possa ser usado com clusters Valkey ou Redis OSS (habilitado no modo de cluster) (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 Valkey ou Redis OSS (modo de cluster desativado) e fizer alterações nos seguintes parâmetros, deverá reinicializar os nós no cluster:

- `activeresharding`
- bancos de dados

Para obter mais informações, consulte [Reinicializando nós](#).

#### Alterações nos parâmetros Valkey ou Redis OSS (modo de cluster ativado)

Se você fizer alterações nos seguintes parâmetros em um cluster Valkey ou Redis OSS (modo de cluster ativado), 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.

## Excluindo um grupo de ElastiCache parâmetros

Você pode excluir um grupo de parâmetros personalizado usando o ElastiCache console AWS CLI, o ou ElastiCache API o.

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 ElastiCache console.

Para excluir um grupo de parâmetros usando o ElastiCache console

1. Faça login no AWS Management Console e abra o ElastiCache console 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 o 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 grupo de myMem14 parâmetros.

## 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 (ElastiCache API)

Para excluir um grupo de parâmetros usando o ElastiCache API, use a `DeleteCacheParameterGroup` ação. 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 `myMem14` parâmetros.

```
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 grupo `myRedde` parâmetros 28.

```
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 motor

### Valkey e Redis OSS

Os parâmetros do Valkey 7.2 são iguais aos parâmetros do Redis OSS 7.

Se você não especificar um grupo de parâmetros para seu OSS cluster Valkey ou Redis, será usado um grupo de parâmetros padrão apropriado à versão do 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 ElastiCache parâmetros](#).

### Tópicos

- [Parâmetros Valkey e Redis OSS](#)
- [Parâmetros específicos do Memcached](#)

## Parâmetros Valkey e Redis OSS

### Tópicos

- [Alterações nos parâmetros Valkey 7.2 e Redis OSS 7](#)
- [Alterações nos parâmetros do Redis OSS 6.x](#)
- [Alterações nos parâmetros do Redis OSS 5.0.3](#)
- [Alterações nos parâmetros do Redis OSS 5.0.0](#)
- [Alterações nos parâmetros do Redis OSS 4.0.10](#)
- [Alterações nos parâmetros do Redis OSS 3.2.10](#)
- [Alterações nos parâmetros do Redis OSS 3.2.6](#)
- [Alterações nos parâmetros do Redis OSS 3.2.4](#)
- [Parâmetros adicionados ao Redis OSS 2.8.24 \(aprimorado\)](#)
- [Parâmetros adicionados ao Redis OSS 2.8.23 \(aprimorado\)](#)
- [Parâmetros adicionados ao Redis OSS 2.8.22 \(aprimorado\)](#)
- [Parâmetros adicionados ao Redis OSS 2.8.21](#)
- [Parâmetros adicionados ao Redis OSS 2.8.19](#)
- [Parâmetros adicionados ao Redis OSS 2.8.6](#)
- [Parâmetros do Redis OSS 2.6.13](#)
- [Parâmetros específicos do tipo de OSS nó Redis](#)

### Alterações nos parâmetros Valkey 7.2 e Redis OSS 7

Família de grupo de parâmetros: redis7

Os grupos de parâmetros padrão do Redis OSS 7 são os seguintes:

- `default.redis7`— Use esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).
- `default.redis7.cluster.on`— Use esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster ativado).

Os parâmetros adicionados no Redis OSS 7 são os seguintes.

Nome	Detalhes	Descrição
<code>cluster-allow-pubsubshard-when-down</code>	<p>Valores permitidos: <code>yes</code>, <code>no</code></p> <p>Padrão: <code>yes</code></p> <p>Tipo: <code>string</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>Quando definido como padrão de <code>yes</code> (sim), permite que os nós forneçam tráfego fragmentado do pubsub enquanto o cluster está inativo, desde que acredite que possui os slots.</p>
<code>cluster-preferred-endpoint-type</code>	<p>Valores permitidos: <code>ip</code>, <code>tls-dynamic</code></p> <p>Padrão: <code>tls-dynamic</code></p> <p>Tipo: <code>string</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>Esse valor controla qual endpoint é retornado para <code>MOVED ASKING /requests</code>, bem como o campo <code>endpoint</code> para <code>e. CLUSTER SLOTS CLUSTER SHARDS</code>. Quando o valor for definido como <code>ip</code>, o nó anunciará seu endereço IP. Quando o valor é definido como <code>tls-dynamic</code>, o nó anunciará um nome de host quando <code>encryption-in-transit</code> estiver habilitado e, caso contrário, um endereço IP.</p>
<code>latency-tracking</code>	<p>Valores permitidos: <code>yes</code>, <code>no</code></p> <p>Padrão: <code>no</code></p> <p>Tipo: <code>string</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>Quando definido como <code>yes</code> (sim), rastreia as latências por comando e permite exportar a distribuição de percentil por meio do comando de estatísticas de latência do <code>INFO</code> e as distribuições de latência cumulativa (histogramas) por meio do comando <code>LATENCY</code>.</p>



Nome	Detalhes	Descrição
hash-max-listpack-entries	<p>Valores permitidos: 0+</p> <p>Padrão: 512</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>O número máximo de entradas de hash para que o conjunto de dados seja compactado.</p>
hash-max-listpack-value	<p>Valores permitidos: 0+</p> <p>Padrão: 64</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>O limite das maiores entradas de hash para que o conjunto de dados seja compactado.</p>
zset-max-listpack-entries	<p>Valores permitidos: 0+</p> <p>Padrão: 128</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>O número máximo de entradas do conjunto classificado para que o conjunto de dados seja compactado.</p>

Nome	Detalhes	Descrição
zset-max-listpack-value	<p>Valores permitidos: 0+</p> <p>Padrão: 64</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	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	Descrição
activeresharding	<p>Permite modificação: no. No Redis OSS 7, esse parâmetro está oculto e ativado por padrão. Para desativá-lo, você precisa criar um <a href="#">caso de suporte</a>.</p>	Permite modificação era Sim.

Os parâmetros removidos no Redis OSS 7 são os seguintes.

Nome	Detalhes	Descrição
hash-max-ziplist-entries	<p>Valores permitidos: 0+</p> <p>Padrão: 512</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p>	Use listpack em vez de ziplist para representar uma pequena codificação de hash

Nome	Detalhes	Descrição
	As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	
hash-max-ziplist-value	Valores permitidos: 0+ Padrão: 64 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Use <code>listpack</code> em vez de <code>ziplist</code> para representar uma pequena codificação de hash
zset-max-ziplist-entries	Valores permitidos: 0+ Padrão: 128 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	Use <code>listpack</code> em vez de <code>ziplist</code> para representar uma pequena codificação de hash.

Nome	Detalhes	Descrição
<code>zset-max-ziplist-value</code>	<p>Valores permitidos: 0+</p> <p>Padrão: 64</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	Use <code>listpack</code> em vez de <code>ziplist</code> para representar uma pequena codificação de hash.
<code>list-max-ziplist-size</code>	<p>Valores permitidos:</p> <p>Padrão: -2</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	O número de entradas permitidas por nó interno da lista.

## Alterações nos 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 esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).
- `default.redis6.x.cluster.on`— Use esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster ativado).

**Note**

Na versão 6.2 OSS do mecanismo Redis, quando a família de nós r6gd foi introduzida para uso com, somente as políticas noeviction [Hierarquização de dados em ElastiCache](#), volatile-lru e allkeys-lru max-memory são suportadas com os tipos de nós r6gd.

Para ter mais informações, consulte [ElastiCache \(RedisOSS\) versão 6.2 \(aprimorada\)](#) e [ElastiCache \(RedisOSS\) versão 6.0 \(aprimorada\)](#).

Os parâmetros adicionados no Redis OSS 6.x são os seguintes.

Detalhes	Descrição	
<code>acl-pubsub-default</code> (added in 6.2)	<p>Valores permitidos: <code>resetchannels</code> , <code>allchannels</code></p> <p>Padrão: <code>allchannels</code></p> <p>Tipo: <code>string</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações entram em vigor: os OSS usuários existentes do Redis associados ao cluster continuarão a ter as permissões existentes. Atualize os usuários ou reinicie o cluster para atualizar os usuários existentes do RedisOSS.</p>	Permissões padrão do canal pubsub para ACL usuários implantados nesse cluster.
<code>cluster-allow-reads-when-do</code>	<p>Padrão: <code>não</code></p> <p>Tipo: <code>string</code></p>	Quando definido como <code>sim</code> , um grupo de replicação do Redis OSS (modo de cluster ativado) continua processando comandos de

Detalhes	Descrição	
wn (added in 6.0)	<p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>leitura mesmo quando um nó não consegue atingir o quórum de primários.</p> <p>Quando definido como o padrão de no (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.</p>
tracking-table-max-keys (added in 6.0)	<p>Padrão: 1.000.000</p> <p>Tipo: número</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>Para auxiliar no armazenamento em cache do lado do cliente, o Redis OSS oferece suporte ao rastreamento de quais clientes acessaram quais chaves.</p> <p>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.</p>

Detalhes	Descrição	
<code>aclog-max-len</code> (added in 6.0)	<p>Padrão: 128</p> <p>Tipo: número</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>Esse valor corresponde ao número máximo de entradas no ACL registro.</p>
<code>active-expire-effort</code> (added in 6.0)	<p>Padrão: 1</p> <p>Tipo: número</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>O Redis OSS exclui as 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.</p> <p>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 desvantagem é uma latência maior CPU e potencialmente maior. Recomendamos um valor de 1, a menos que você esteja vendo um alto uso de memória e possa tolerar um aumento na CPU utilização.</p>

Detalhes	Descrição	
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	Descrição
lua-replique-commands	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 nos 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 esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).
- `default.redis5.0.cluster.on`— Use esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster ativado).



## Parâmetros adicionados no Redis 5.0.3 OSS

Nome	Detalhes	Descrição
rename-commands	<p>Padrão: nenhum</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	<p>Uma lista separada por espaços de comandos do Redis renomeados. OSS Veja a seguir uma lista restrita de comandos disponíveis para renomear:</p> <p>APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR DECRBY DEL DISCARD DUMP ECHO EVAL EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS GEORADIUSBYMEMBER GET GETBIT GETRANGE GETSET HDEL HEXISTS HGET HGETALL HINCRBY HINCRBYFLOAT HKEYS HLEN HMGET HMSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPOS LPUSH LPUSHX LRANGE LREM LSET LTRIM MEMORY MGET MONITOR MOVE MSET MSETNX MULTI OBJECT PERSIST PEXPIRE PEXPIREAT PFADD PFCOUNT PFMERGE PING PSETEX PSUBSCRIBE PUBSUB PTTL PUBLISH PUNSUBSCRIBE RANDOMKEY READONLY READWRITE RENAME RENAMENX RESTORE ROLE RPOP RPOPLUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET SETBIT SETEX SETNX SETRANGE SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP</p>

Nome	Detalhes	Descrição
		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 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 \(RedisOSS\) versão 5.0.6 \(aprimorada\)](#).

Alterações nos 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 esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).
- `default.redis5.0.cluster.on`— Use esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster ativado).

## Parâmetros adicionados no Redis 5.0 OSS

Nome	Detalhes	Descrição
<code>stream-node-max-bytes</code>	<p>Valores permitidos: 0+</p> <p>Padrão: 4096</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entram em vigor: imediatamente</p>	<p>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.</p>
<code>stream-node-max-entries</code>	<p>Valores permitidos: 0+</p> <p>Padrão: 100</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entram em vigor: imediatamente</p>	<p>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</p>
<code>active-defrag-max-scan-fields</code>	<p>Valores permitidos: 1 a 1000000</p> <p>Padrão: 1000</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações entram em vigor: imediatamente</p>	<p>Número máximo de conjunto/hash/zset/lista de campos que serão processados a partir da verificação do dicionário principal</p>

Nome	Detalhes	Descrição
lua-replicate-commands	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
replica-ignore-maxmemory	Padrão: sim Tipo: booliano Modificável: não	Determina se a réplica ignora a configuração maxmemory ao não remover itens independentes do primário

O Redis OSS renomeou vários parâmetros na versão 5.0 do motor 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 5.0 OSS

Nome	Detalhes	Descrição
replica-lazy-flush	Padrão: sim Tipo: booliano Modificável: não Nome anterior: slave-lazy-flush	Realiza um flushDB assíncrono durante a sincronização da réplica.
client-output-buffer	Padrão: para valores, consulte <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a>	Para réplicas de OSS leitura do Redis: se o buffer de saída de um cliente atingir o número

Nome	Detalhes	Descrição
er-limit-replica-hard-limit	<p>Tipo: inteiro</p> <p>Modificável: não</p> <p>Nome anterior: client-output-buffer-limit - slave-hard-limit</p>	especificado de bytes, o cliente será desconectado.
client-output-buffer-limit-replica-soft-limit	<p>Padrão: para valores, consulte <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a></p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>Nome anterior: client-output-buffer-limit - slave-soft-limit</p>	Para réplicas de OSS leitura do Redis: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado, mas somente se essa condição persistir . client-output-buffer-limit-replica-soft-seconds
client-output-buffer-limit-replica-soft-seconds	<p>Padrão: 60</p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>Nome anterior: client-output-buffer-limit - slave-soft-seconds</p>	Para réplicas de OSS leitura do Redis: se o buffer de saída de um cliente permanecer em client-output-buffer-limit-replica-soft-limit bytes por mais do que esse número de segundos, o cliente será desconectado.
replica-allow-chaining	<p>Padrão: não</p> <p>Tipo: string</p> <p>Modificável: não</p> <p>Nome anterior: slave-allow-chaining</p>	Determina se uma réplica de leitura no Redis OSS pode ter suas próprias réplicas de leitura.

Nome	Detalhes	Descrição
<code>min-replicas-to-write</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>Nome anterior: min-slaves-to-write</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p> <p>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.</p>
<code>min-replicas-max-lag</code>	<p>Padrão: 10</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>Nome anterior: min-slaves-max-lag</p> <p>As alterações terão efeito: imediatamente</p>	<p>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-replicas-to-write</code>, a primária deixará de aceitar gravações nesse momento.</p> <p>Se esse parâmetro ou <code>min-replicas-to-write</code> for 0, o nó primário sempre aceitará solicitações de gravação, mesmo que nenhuma réplica esteja disponível.</p>
<code>close-on-replica-write</code>	<p>Padrão: sim</p> <p>Tipo: booliano</p> <p>Modificável: sim</p> <p>Nome anterior: close-on-slave-write</p> <p>As alterações terão efeito: imediatamente</p>	<p>Se ativado, os clientes que tentarem gravar em uma réplica somente leitura serão desconectados.</p>

## Parâmetros removidos no Redis 5.0 OSS

Nome	Detalhes	Descrição
repl-timeout	Padrão: 60 Modificável: não	O parâmetro não está disponível nesta versão.

## Alterações nos 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 esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster desativado).
- `default.redis4.0.cluster.on`— Use esse grupo de parâmetros, ou um derivado dele, para clusters e grupos de replicação Valkey ou Redis OSS (modo de cluster ativado).

## Parâmetros alterados no Redis 4.0.10 OSS

Nome	Detalhes	Descrição
maxmemory-policy	Valores permitidos: allkeys-lru , volatile-lru , <b>allkeys-lfu</b> , <b>volatile-lfu</b> , allkeys-random , volatile-random , volatile-ttl , noeviction Padrão: volatile-lru Tipo: string Modificável: sim	<code>maxmemory-policy</code> foi adicionado na versão 2.6.13. Na versão 4.0.10, dois novos valores permitidos são adicionados: <code>allkeys-lfu</code> , que despejará qualquer chave usando aproximada e <code>LFUvolatile-lfu</code> , que despejará usando aproximada entre as chaves com uma expiração definida LFU. Na versão 6.2, quando a família de nós <code>r6gd</code> foi introduzida para uso com classificação de dados em níveis, somente as políticas <code>noeviction</code> , <code>volatile-lru</code> e <code>allkeys-lru</code> são compatíveis com os tipos de nó <code>r6gd</code> .

Nome	Detalhes	Descrição
	As alterações ocorrem: imediatamente	

## Parâmetros adicionados no Redis 4.0.10 OSS

Nome	Detalhes	Descrição
Parâmetros de exclusão assíncrona		
<code>lazyfree-lazy- eviction</code>	Valores permitidos: sim/não  Padrão: não  Tipo: booliano  Modificável: sim  As alterações ocorrem: imediatamente	Realiza uma exclusão assíncrona em remoções.
<code>lazyfree-lazy-expire</code>	Valores permitidos: sim/não  Padrão: não  Tipo: booliano  Modificável: sim  As alterações ocorrem: imediatamente	Realiza uma exclusão assíncrona em chaves expiradas.
<code>lazyfree-lazy-server-del</code>	Valores permitidos: sim/não  Padrão: não  Tipo: booliano	Realiza uma exclusão assíncrona para comandos que atualizam valores.



Nome	Detalhes	Descrição
	Modificável: sim	
	As alterações ocorrem: imediatamente	
<code>slave-lazy-flush</code>	Valores permitidos: N/D	Realiza um flushDB assíncrono durante a sincronização subordinada.
	Padrão: não	
	Tipo: booleano	
	Modificável: não	
	As alterações ocorrem: N/D	
LFUparâmetros		
<code>lfu-log-factor</code>	Valores permitidos: qualquer inteiro > 0	Defina o fator de log, que determina o número de hits de chave para saturar o contador de chaves.
	Padrão: 10	
	Tipo: inteiro	
	Modificável: sim	
	As alterações ocorrem: imediatamente	

Nome	Detalhes	Descrição
<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		
<code>activedefrag</code>	Valores permitidos: sim/não  Padrão: não  Tipo: booleano  Modificável: sim  As alterações ocorrem: imediatamente	Desfragmentação ativa habilitada.
<code>active-defrag-ignore-bytes</code>	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	Descrição
<code>active-defrag-threshold-lower</code>	Valores permitidos: 1-100 Padrão: 10 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Porcentagem mínima de fragmentação para iniciar a desfragmentação ativa.
<code>active-defrag-threshold-upper</code>	Valores permitidos: 1-100 Padrão: 100 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Porcentagem máxima de fragmentação em que usamos o esforço máximo.
<code>active-defrag-cycle-min</code>	Valores permitidos: 1-75 Padrão: 25 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Esforço mínimo para desfragmentar em CPU porcentagem.

Nome	Detalhes	Descrição
<code>active-defrag-cycle-max</code>	Valores permitidos: 1-75 Padrão: 75 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Esforço máximo para desfragmentar em porcentagem em CPU
Parâmetros de buffer de saída do cliente		
<code>client-query-buffer-limit</code>	Valores permitidos: 1048576-1073741824 Padrão: 1073741824 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Tamanho máximo de um único buffer de consulta do cliente.
<code>proto-max-bulk-len</code>	Valores permitidos: 1048576-536870912 Padrão: 536870912 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Tamanho máximo de uma única solicitação de elemento.

## Alterações nos parâmetros do Redis OSS 3.2.10

Família de grupo de parâmetros: redis3.2

ElastiCache (RedisOSS) 3.2.10 não há suporte para parâmetros adicionais.

## Alterações nos parâmetros do Redis OSS 3.2.6

Família de grupo de parâmetros: redis3.2

Para o Redis OSS 3.2.6, não há suporte a parâmetros adicionais.

## Alterações nos parâmetros do Redis OSS 3.2.4

Família de grupo de parâmetros: redis3.2

A partir do Redis OSS 3.2.4, há dois grupos de parâmetros padrão.

- `default.redis3.2`— Ao executar o Redis OSS 3.2.4, especifique esse grupo de parâmetros ou um derivado dele, se quiser criar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) e ainda usar os recursos adicionais do Redis 3.2.4. OSS
- `default.redis3.2.cluster.on`— Especifique esse grupo de parâmetros ou um derivado dele, quando quiser criar um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado).

## Tópicos

- [Novos parâmetros para o Redis 3.2.4 OSS](#)
- [Parâmetros alterados no Redis OSS 3.2.4 \(aprimorado\)](#)

## Novos parâmetros para o Redis 3.2.4 OSS

Família de grupo de parâmetros: redis3.2

Para o Redis OSS 3.2.4, os seguintes parâmetros adicionais são suportados.

Nome	Detalhes	Descrição
<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	Descrição
	Modificável: não	<p>fixo ou um número máximo de elementos. Para um tamanho máximo fixo, use de -5 a -1, significando:</p> <ul style="list-style-type: none"><li>• -5: tamanho máximo: 64 Kb - não recomendado para workloads normais</li><li>• -4: tamanho máximo: 32 Kb - não recomendado</li><li>• -3: tamanho máximo: 16 Kb - não recomendado</li><li>• -2: tamanho máximo: 8 Kb - recomendado</li><li>• -1: tamanho máximo: 4 Kb - recomendado</li><li>• Números positivos significam armazenar exatamente o número de elementos por nó de lista.</li></ul>

Nome	Detalhes	Descrição
<code>list-compress-depth</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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:</p> <ul style="list-style-type: none"><li>• 0: desabilitar toda a compactação.</li><li>• 1: começar a compactar com o 1º nó de início e final.  [início] -&gt; nó-&gt; nó -&gt; ...-&gt; nó -&gt; [final]  Todos os nós, exceto [início] e [final] são compactados.</li><li>• 2: começar a compactar com o 2º nó de início e final.  [início] -&gt; [próximo] -&gt; nó-&gt; nó -&gt; ...-&gt; nó -&gt; [anterior] -&gt; [final]  [início], [próximo], [anterior], [final] não são compactados. Todos os outros nós são compactados.</li><li>• Etc.</li></ul>

Nome	Detalhes	Descrição
<code>cluster-enabled</code>	<p>Padrão: não/sim*</p> <p>Tipo: string</p> <p>Modificável: não</p>	<p>Indica se esse é um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) no modo de cluster (sim) ou um grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) em modo não cluster (não). Os grupos de replicação Valkey ou Redis OSS (modo de cluster ativado) no modo cluster podem particionar seus dados em até 500 grupos de nós.</p> <p>* Redis OSS 3.2. x tem dois grupos de parâmetros padrão.</p> <ul style="list-style-type: none"><li>• <code>default.redis3.2</code> – valor padrão no.</li><li>• <code>default.redis3.2.cluster.on</code> – valor padrão yes.</li></ul> <p>.</p>



Nome	Detalhes	Descrição
<code>cluster-require-full-coverage</code>	<p>Padrão: não</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>Quando definidos com <code>yes</code>, os nós Valkey ou Redis OSS (modo de cluster ativado) no modo cluster param de aceitar consultas se detectarem que há pelo menos um slot de hash descoberto (nenhum nó disponível o está servindo). 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.</p> <p>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 <code>no</code>.</p>

Nome	Detalhes	Descrição
hll-spars e-max-byt es	<p>Padrão: 3000</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>HyperLogLog limite de bytes de representação 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.</p> <p>Não é recomendado um valor superior a 16000, porque, nesse ponto, a representação densa é mais eficiente em termos de memória.</p> <p>Recomendamos um valor de cerca de 3000 para obter os benefícios da codificação eficiente em termos de espaço sem diminuir PFADD muito a velocidade, que é <math>O(N)</math> com a codificação esparsa. O valor pode ser aumentado para ~ 10000 quando não CPU 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.</p>

Nome	Detalhes	Descrição
<code>reserved-memory-percent</code>	<p>Padrão: 25</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>A porcentagem da memória de um nó reservada para o uso não de dados. Por padrão, a área de cobertura de OSS dados do Redis aumenta 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 reservar parte da memória disponível para OSS fins que não sejam do Redis para ajudar a reduzir a quantidade de paginação.</p> <p>Esse parâmetro é específico e não faz parte da OSS distribuição padrão do Redis. ElastiCache</p> <p>Para ter mais informações, consulte <code>reserved-memory</code> e <a href="#">Gerenciando a memória reservada para Valkey e Redis OSS</a>.</p>

### Parâmetros alterados no Redis OSS 3.2.4 (aprimorado)

Família de grupo de parâmetros: `redis3.2`

Para o Redis OSS 3.2.4, os seguintes parâmetros foram alterados.

Nome	Detalhes	Alteração
<code>activeresharding</code>	Modificável: Sim se o grupo de parâmetros não estiver associado a nenhum cluster de cache. Do contrário, não.	Modificável era Não.
<code>databases</code>	Modificável: Sim se o grupo de parâmetros não estiver	Modificável era Não.

Nome	Detalhes	Alteração
	associado a nenhum cluster de cache. Do contrário, não.	
appendonly	Padrão: desativado Modificável: não	Se você quiser fazer o upgrade de uma OSS versão anterior do Redis, primeiro <code>appendonly</code> desative.
appendfsync	Padrão: desativado Modificável: não	Se você quiser fazer o upgrade de uma OSS versão anterior do Redis, primeiro <code>appendfsync</code> desative.
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 ao Redis OSS 2.8.24 (aprimorado)

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.24, não há suporte a parâmetros adicionais.

Parâmetros adicionados ao Redis OSS 2.8.23 (aprimorado)

Família de grupo de parâmetros: redis2.8

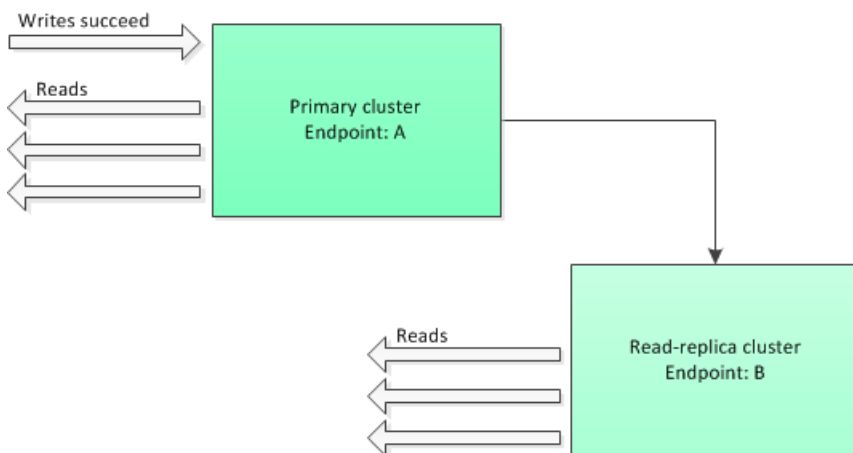
Para o Redis OSS 2.8.23, o seguinte parâmetro adicional é suportado.

Nome	Detalhes	Descrição
close-on-slave-write	<p>Padrão: sim</p> <p>Tipo: string (sim/não)</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	Se ativado, os clientes que tentarem gravar em uma réplica somente leitura serão desconectados.

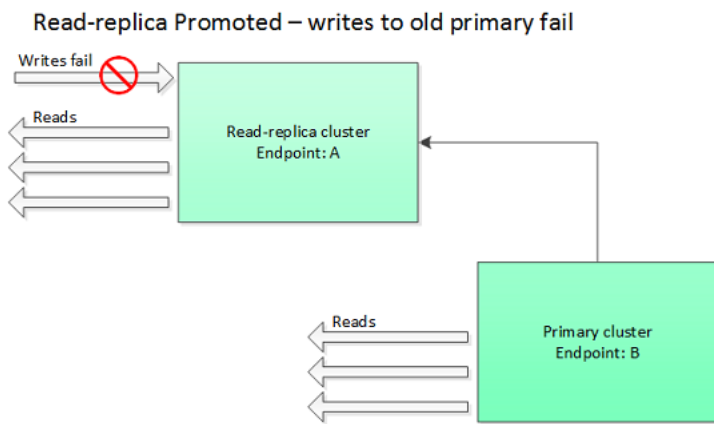
### Como close-on-slave-write funciona

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.

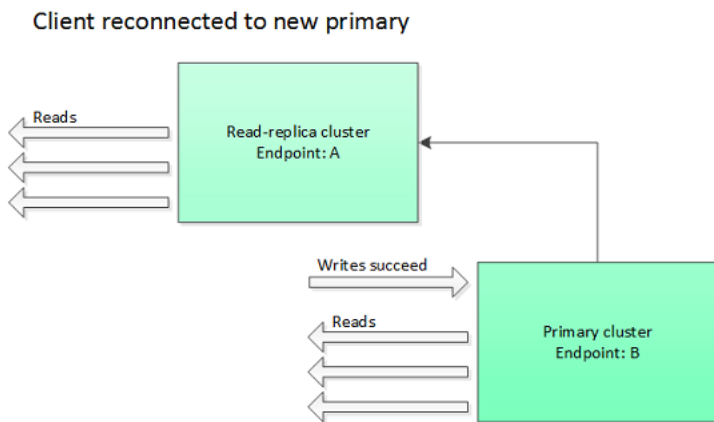
#### Before read-replica promotion



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`.



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. A lógica do aplicativo deve detectar a desconexão, verificar a DNS tabela e se reconectar ao endpoint primário, que agora seria o endpoint B.



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 ao Redis OSS 2.8.22 (aprimorado)

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.22, não há suporte a parâmetros adicionais.

### Important

- A partir da OSS versão 2.8.22 do Redis, `repl-backlog-size` aplica-se ao cluster primário e aos clusters de réplica.
- A partir da OSS versão 2.8.22 do Redis, o `repl-timeout` parâmetro não é suportado. 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 ao Redis OSS 2.8.21

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.21, não há suporte para parâmetros adicionais.

## Parâmetros adicionados ao Redis OSS 2.8.19

Família de grupo de parâmetros: redis2.8

Para o Redis OSS 2.8.19, não há suporte a parâmetros adicionais.

## Parâmetros adicionados ao Redis OSS 2.8.6

Família de grupo de parâmetros: redis2.8


Para o Redis OSS 2.8.6, os seguintes parâmetros adicionais são suportados.

Nome	Detalhes	Descrição
<code>min-slaves-max-lag</code>	<p>Padrão: 10</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p> <p>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.</p>
<code>min-slaves-to-write</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p> <p>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.</p>
<code>notify-keyspace-events</code>	<p>Padrão: (uma string vazia)</p>	<p>Os tipos de eventos de keyspace sobre os quais o Redis OSS pode</p>



Nome	Detalhes	Descrição
	<p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>notificar os clientes. Cada tipo de evento é representado por uma única letra:</p> <ul style="list-style-type: none"> <li>• K: eventos de espaço de chaves, publicados com um prefixo de <code>__keyspace@&lt;db&gt;__</code></li> <li>• E: eventos-chave, publicados com um prefixo de <code>__keyevent@&lt;db&gt;__</code></li> <li>• g — Comandos genéricos e não específicos DEL, como, EXPIRE, RENAME, etc.</li> <li>• \$ — Comandos de string</li> <li>• l — Listar comandos</li> <li>• s — Definir comandos</li> <li>• h — Comandos de hash</li> <li>• z — Comandos de conjuntos classificados</li> <li>• x — Eventos expirados (eventos gerados sempre que uma chave expira)</li> <li>• e — Eventos removidos (eventos gerados quando uma chave é removida para <code>maxmemory</code>)</li> </ul>

Nome	Detalhes	Descrição
		<ul style="list-style-type: none"><li data-bbox="1008 216 1446 279">• A — Um alias para g\$shzxe</li></ul> <p data-bbox="1008 384 1511 611">Você pode ter qualquer combinação o desses tipos de eventos. Por exemplo, AKEsignifica que o Redis OSS pode publicar notificações de todos os tipos de eventos.</p> <p data-bbox="1008 653 1479 831">Não use nenhum caractere diferente dos listados acima. Se isso for feito, mensagens de erro serão geradas.</p> <p data-bbox="1008 873 1503 1100">Por padrão, esse parâmetro é definido como uma string vazia, o que significa que a notificação de eventos de espaço de chaves está desabilitada.</p>


Nome	Detalhes	Descrição
<code>repl-backlog-size</code>	<p>Padrão: 1048576</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p> <p>O valor mínimo para esse parâmetro é 16384.</p> <div data-bbox="1008 957 1507 1270"><p> <b>Note</b></p><p>A partir do Redis OSS 2.8.22, esse parâmetro se aplica ao cluster primário e às réplicas de leitura.</p></div>


Nome	Detalhes	Descrição
<code>repl-backlog-ttl</code>	<p>Padrão: 3600</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p> <p>Se esse parâmetro estiver definido como 0, o buffer de backlog nunca será liberado.</p>
<code>repl-timeout</code>	<p>Padrão: 60</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>Representa o período de tempo limite, em segundos, para:</p> <ul style="list-style-type: none"><li>• Transferência de dados em massa durante a sincronização, sob a perspectiva da réplica de leitura</li><li>• Tempo limite do nó primário, sob a perspectiva da réplica</li><li>• Tempo limite da réplica, sob a perspectiva do nó primário</li></ul>

## 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	Descrição
activereshashing	<p>Padrão: sim</p> <p>Tipo: string (sim/não)</p> <p>Modificável: sim</p> <p>As alterações ocorrem: na criação</p>	<p>Determina se deseja o recurso de rehashing ativo do Redis deve ou não ser habilitado. A tabela de hash principal é refeita dez vezes por segundo; cada operação de rehash consome 1 milissegundo de tempo. CPU</p> <p>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.</p>
appendonly	<p>Padrão: não</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>Ativa ou desativa o recurso de anexar somente arquivos do Redis (). AOF captura todos OSS os comandos do Redis que alteram dados no cache e é usado para se recuperar de certas falhas do nó.</p> <p>O valor padrão é não, ou seja, AOF está desativado. Defina esse parâmetro como sim para habilitar AOF.</p> <p>Para obter mais informações, consulte <a href="#">Atenuar falhas</a>.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Anexar somente arquivos (AOF) não é compatível com os nós cache.t1.micro e cache.t2.*. Para nós desse tipo, o</p> </div>

Nome	Detalhes	Descrição
		<p>valor do parâmetro <code>appendonly</code> é ignorado.</p> <p> <b>Note</b> Para grupos de replicação Multi-AZ, não AOF é permitido.</p>
<code>appendfsync</code>	<p>Padrão: <code>everysec</code></p> <p>Tipo: <code>string</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações terão efeito: imediatamente</p>	<p>Quando <code>appendonly</code> definido como <code>sim</code>, controla a frequência com que o buffer AOF de saída é gravado no disco:</p> <ul style="list-style-type: none"> <li><code>no</code> — o buffer é liberado no disco, conforme necessário.</li> <li><code>everysec</code>: o buffer é liberado uma vez por segundo. Esse é o padrão.</li> <li><code>always</code> — o buffer é liberado sempre que os dados no cluster são modificados.</li> <li>O <code>Appendfsync</code> não é compatível com as versões 2.8.22 e posteriores.</li> </ul>
<code>client-output-buffer-limit-normal-hard-limit</code>	<p>Padrão: <code>0</code></p> <p>Tipo: <code>inteiro</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações terão efeito: imediatamente</p>	<p>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).</p>

Nome	Detalhes	Descrição
<code>client-output-buffer-limit-normal-soft-limit</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	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>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	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>	<p>Padrão: 33554432</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	Para clientes de OSS publicação/assinatura do Redis: 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>	<p>Padrão: 8388608</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	Para clientes de OSS publicação/assinatura do Redis: 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	Descrição
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	<p>Padrão: 60</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	Para clientes de OSS publicação/assinatura do Redis: 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>	<p>Padrão: para valores, consulte <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a></p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	Para réplicas de OSS leitura do Redis: 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>	<p>Padrão: para valores, consulte <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a></p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	Para réplicas de OSS leitura do Redis: 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-slave-soft-seconds</code>
<code>client-output-buffer-limit-slave-soft-seconds</code>	<p>Padrão: 60</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	Para réplicas de OSS leitura do Redis: se o buffer de saída de um cliente permanecer em <code>client-output-buffer-limit-slave-soft-limit</code> bytes por mais do que esse número de segundos, o cliente será desconectado.




Nome	Detalhes	Descrição
databases	<p>Padrão: 16</p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>As alterações ocorrem: na criação</p>	<p>O número de partições lógicas em que os bancos de dados estão divididos. Recomendamos manter esse valor baixo.</p> <p>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.</p>
hash-max-ziplist-entries	<p>Padrão: 512</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>
hash-max-ziplist-value	<p>Padrão: 64</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>
list-max-ziplist-entries	<p>Padrão: 512</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>

Nome	Detalhes	Descrição
<code>list-max-ziplist-value</code>	<p>Padrão: 64</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>
<code>lua-time-limit</code>	<p>Padrão: 5000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	<p>O tempo máximo de execução de um script Lua, em milissegundos, antes de ElastiCache agir para interromper o script.</p> <p>Se <code>lua-time-limit</code> for excedido, todos os OSS comandos do Redis retornarão um erro no formato <code>____ - BUSY</code>. Como esse estado pode causar interferência em muitas OSS operações essenciais do Redis, primeiro ElastiCache emitirá um <code>SCRIPTKILL</code> comando. Se isso não for bem-sucedido, ElastiCache reiniciará o Redis à força. OSS</p>
<code>maxclients</code> Esse valor se aplica a todos os tipos de instâncias, exceto às instâncias explicitamente especificadas	<p>Padrão: 65000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>t2.medium Padrão: 20000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>t2.small Padrão: 20000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	<p>O número máximo de clientes que podem ser conectados ao mesmo tempo.</p>

Nome	Detalhes	Descrição
	<p>t2.micro Padrão: 20000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	
	<p>t4g.micro Padrão: 20000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	
	<p>t3.medium Padrão: 46000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	
	<p>t3.small Padrão: 46000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	
	<p>t3.micro Padrão: 20000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	
maxmemory-policy	<p>Padrão: volatile-lru</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>A política de remoção de chaves quando o uso máximo da memória é atingido.</p> <p>Os valores válidos são: volatile-lru   allkeys-lru   volatile-random   allkeys-random   volatile-ttl   noeviction</p> <p>Para obter mais informações, consulte <a href="#">Usando Valkey ou Redis OSS como cache</a>. LRU</p>

Nome	Detalhes	Descrição
<code>maxmemory-samples</code>	<p>Padrão: 3</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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 teclas e usa a que foi usada menos recentemente.</p>
<code>reserved-memory</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>A memória total, em bytes, reservada para o uso de não dados. Por padrão, o OSS nó Redis crescerá até consumir o nó <code>maxmemory</code> (consulte <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a>). Se isso ocorrer, o desempenho do nó provavelmente sofrerá devido a uma paginação de memória excessiva. Ao reservar memória, você pode reservar parte da memória disponível para OSS fins que não sejam do Redis para ajudar a reduzir a quantidade de paginação.</p> <p>Esse parâmetro é específico e não faz parte da OSS distribuição padrão do Redis. ElastiCache</p> <p>Para ter mais informações, consulte <code>reserved-memory-percent</code> e <a href="#">Gerenciando a memória reservada para Valkey e Redis OSS</a>.</p>
<code>set-max-intset-entries</code>	<p>Padrão: 512</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>

Nome	Detalhes	Descrição
slave-allow-chaining	Padrão: não Tipo: string Modificável: não	Determina se uma réplica de leitura no Redis OSS pode ter suas próprias réplicas de leitura.
slowlog-log-slower-than	Padrão: 10000 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O tempo máximo de execução, em microssegundos, dos comandos a serem registrados pelo recurso Redis OSS Slow Log.
slowlog-max-len	Padrão: 128 Tipo: inteiro Modificável: sim As alterações terão efeito: imediatamente	O comprimento máximo do Redis OSS Slow Log.
tcp-keepalive	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.  <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> <b>Important</b></p><p>Alguns aspectos desse parâmetro foram alterados na OSS versão 3.2.4 do Redis. Consulte <a href="#">Parâmetros alterados no Redis OSS 3.2.4 (aprimorado)</a>.</p></div>

Nome	Detalhes	Descrição
timeout	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>O número de segundos que um nó espera antes do tempo limite. Os valores são:</p> <ul style="list-style-type: none"> <li>• 0 – nunca desconectar um cliente ocioso.</li> <li>• 1-19 – valores inválidos.</li> <li>• <math>\geq 20</math> – o número de segundos que um nó espera antes de desconectar um cliente ocioso.</li> </ul>
zset-max-ziplist-entries	<p>Padrão: 128</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>
zset-max-ziplist-value	<p>Padrão: 64</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p>

### Note

Se você não especificar um grupo de parâmetros para seu cluster 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 do tipo de OSS nó Redis

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 instância 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 Redis. OSS AOF

Variáveis de OSS configuração do Redis `appendonly` e `appendfsync` são compatíveis com a OSS versão 2.8.22 e posterior do Redis.

## 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 ElastiCache parâmetros](#).

### 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 alterações, você não poderá ativar/desativar `lru_crawler` no runtime por meio de comandos. Ative/desative `lru_crawler` modificando seu grupo de parâmetros personalizados.

## 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	Descrição
<code>no_modern</code>	<p>Padrão: 1</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>Valores permitidos: 0,1</p> <p>As alterações terão efeito: na execução</p>	<p>Um alias para desativar <code>slab_reassign</code> comandos <code>lru_maintainer_thread</code>, <code>lru_segmented</code>, e <code>maxconns_fast</code></p> <p>Ao usar o Memcached 1.5 e superior, <code>no_modern</code> também</p>

Nome	Detalhes	Descrição
		<p>define o <code>hash_algorithm</code> como <code>jenkins</code>.</p> <p>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 desativado, <code>inline_ascii_reponse</code> estará desativado. Do mecanismo Memcached 1.5.16 em diante, o <code>inline_ascii_response</code> parâmetro não se aplica mais, portanto, <code>no_modern</code> ser ativado ou desativado não tem efeito sobre <code>inline_ascii_reponse</code>.</p> <p>Se <code>no_modern</code> estiver desativado <code>slab_reassign</code>, então <code>lru_maintainer_thread</code>, <code>lru_segmented</code>, e <code>maxconns_fast</code> WILL seja ativado. Como <code>hash_algorithm</code> os parâmetros <code>slab_automove</code> e não são SWITCH parâmetros, sua configuração é baseada nas configurações do grupo de parâmetros.</p> <p>Se quiser desabilitar <code>no_modern</code> e reverter <code>paramodern</code>, você deve configurar um grupo de</p>

Nome	Detalhes	Descrição
		<p>parâmetros personalizado para desabilitar esse parâmetro e, em seguida, reinicializar para que essas alterações entrem em vigor.</p> <div data-bbox="1008 430 1507 1459"><p> <b>Note</b></p><p>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.</p></div>

Nome	Detalhes	Descrição
<code>inline_ascii_resp</code>	<p>Padrão: 0</p> <p>Tipo: booliano</p> <p>Modificável: sim</p> <p>Valores permitidos: 0,1</p> <p>As alterações terão efeito: na execução</p>	Armazena números da resposta VALUE, dentro de um item, usando até 24 bytes. Pequena desaceleração para ASCIIget, faster sets.

Para a versão 1.5.10 do Memcached, os parâmetros seguintes foram removidos.

Nome	Detalhes	Descrição
<code>expirezero_does_no_t_evict</code>	<p>Padrão: 0</p> <p>Tipo: booliano</p> <p>Modificável: sim</p> <p>Valores permitidos: 0,1</p> <p>As alterações terão efeito: na execução</p>	Não é mais compatível nesta versão.
<code>modern</code>	<p>Padrão: 1</p> <p>Tipo: booliano</p> <p>Modificável: sim (requer reinicialização se definido para <code>no_modern</code> )</p>	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	Descrição
	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	Descrição
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	Descrição
	<p>As alterações terão efeito: imediatamente</p> <p>Logs podem ser descartados se o usuário atingir seus limites de <code>watcher_logbuf_size</code> e <code>worker_logbuf_size</code>.</p>	
<code>idle_timeout</code>	<p>Padrão: 0 (desabilitado)</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>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.</p>
<code>track_sizes</code>	<p>Padrão: desabilitado</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>Mostra os tamanhos que cada grupo de slabs consumiu.</p> <p>Habilitar <code>track_sizes</code> permite que você execute <code>stats sizes</code> sem a necessidade de executar <code>stats sizes_enable</code>.</p>

Nome	Detalhes	Descrição
<code>watcher_logbuf_size</code>	<p>Padrão: 256 (KB)</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>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.</p>
<code>worker_logbuf_size</code>	<p>Padrão: 64 (KB)</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>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.</p>
<code>slab_chunk_max</code>	<p>Padrão: 524288 (bytes)</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>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.</p>



Nome	Detalhes	Descrição
<code>lru_crawler metadump [all 1 2 3]</code>	<p>Padrão: desabilitado</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>se <code>lru_crawler</code> estiver habilitado, esse comando despejará todas as chaves.</p> <p><code>all 1 2 3</code> - todos os slab, ou especifique um número específico de slab</p>


### 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	Descrição
<code>disable_flush_all</code>	<p>Padrão: 0 (desabilitado)</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>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.</p> <p>Valores: 0, 1 (o usuário pode fazer <code>flush_all</code> quando o valor é 0).</p>
<code>hash_algorithm</code>	<p>Padrão: jenkins</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>O algoritmo de hash a ser usado. Valores permitidos: <code>murmur3</code> e <code>jenkins</code>.</p>
<code>lru_crawler</code>	<p>Padrão: 0 (desabilitado)</p>	

Nome	Detalhes	Descrição
	<p>Tipo: booliano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p> <div data-bbox="651 541 971 1381" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>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.</p> </div>	<p>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.</p> <p>Para habilitar temporariamente, execute <code>lru_crawler enable</code> na linha de comando.</p> <p><code>lru_crawler 1,3,5</code> rastreia as classes de slab 1, 3 e 5, procurando itens expirados para adicionar à lista livre.</p> <p>Valores: 0,1</p> <div data-bbox="1008 1052 1507 1843" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>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 <a href="#">Modificando um grupo de ElastiCache parâmetros</a>.</p> </div>

Nome	Detalhes	Descrição
<code>lru_maintainer</code>	<p>Padrão: 0 (desabilitado)</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>Uma linha de fundo que embaralha os itens entre eles à LRUs medida que as capacidades são atingidas. Valores: 0, 1.</p>
<code>expirezero_does_not_evict</code>	<p>Padrão: 0 (desabilitado)</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>Quando usado com <code>lru_maintainer</code>, faz com que itens com um tempo de expiração de 0 não possam ser removidos.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Warning</b></p> <p>Isso pode lotar a memória disponível para outros itens removíveis.</p> </div> <p>Pode ser configurado para desconsiderar <code>lru_maintainer</code>.</p>

## 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	Descrição
<code>config_max</code>	O número máximo de entradas de ElastiCache configuração.
<code>config_size_max</code>	O tamanho máximo das entradas de configuração, em bytes.
<code>hashpower_init</code>	O tamanho inicial da tabela de ElastiCache hash, expresso como uma potência de dois. O padrão é 16 ( $2^{16}$ ) ou 65536 chaves.
<code>maxconns_fast</code>	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	Descrição
slab_automove	<p>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, ElastiCache adota uma abordagem lenta e conservadora para mover placas 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.)</p>



Nome	Descrição
<code>slab_reassign</code>	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	Descrição
<code>backlog_queue_limit</code>	Padrão: 1024 Tipo: inteiro Modificável: não	O limite de fila de backlog.

Nome	Detalhes	Descrição
<code>binding_protocol</code>	<p>Padrão: automático</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	<p>O protocolo de associação.</p> <p>Os valores permitidos são: <code>ascii</code> e <code>auto</code>.</p> <p>Para obter orientação sobre como modificar o valor de <code>binding_protocol</code>, consulte <a href="#">Modificando um grupo de ElastiCache parâmetros</a>.</p>
<code>cas_disabled</code>	<p>Padrão: 0 (false)</p> <p>Tipo: booliano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	<p>Se 1 (verdadeiro), as operações <code>check</code> e <code>set</code> (CAS) serão desativadas e os itens armazenados consumirão 8 bytes a menos do que com CAS habilitado.</p>
<code>chunk_size</code>	<p>Padrão: 48</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	<p>O valor mínimo, em bytes, de espaço para alocar para a chave, o valor e os sinalizadores do menor item.</p>
<code>chunk_size_growth_factor</code>	<p>Padrão: 1.25</p> <p>Tipo: flutuante</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	<p>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.</p>

Nome	Detalhes	Descrição
<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	Descrição
<code>maximize_core_file_limit</code>	<p>Padrão: 0 (false)</p> <p>Tipo: booleano</p> <p>Permite modificação:</p> <p>As alterações terão efeito: após a reinicialização</p>	Se 1 (verdadeiro), ElastiCache maximizará o limite do arquivo principal.
<code>memcached_connections_overhead</code>	<p>Padrão: 100</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	A quantidade de memória a ser reservada para conexões Memcached e outras sobrecargas diversas. Para obter informações sobre esse parâmetro, consulte <a href="#">Sobrecarga de conexões do Memcached</a> .
<code>requests_per_event</code>	<p>Padrão: 20</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	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 (AmazonVPC).

## Dimensionamento ElastiCache

Você pode escalar seu ElastiCache cache para atender às suas necessidades. Caches sem servidor e clusters autoprojeto 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 memória CPU 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 que estão sendo 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 em ECPU /second do seu cache para controlar os custos do cache. Isso vai garantir que o uso do cache jamais exceda o máximo configurado.

Se você definir um máximo de escalabilidade, seu aplicativo poderá ter um desempenho de cache reduzido quando o cache atingir 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 LRU lógica. Se não houver dados que possam ser removidos, as solicitações para gravar dados adicionais receberão uma mensagem de erro Out Of Memory (OOM). Quando você define um máximo de ECPU /segundo 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, CLI, ou API ElastiCache O Serverless atualiza o ECPUs /segundo disponível no cache em 60 minutos e envia uma notificação de evento quando a atualização do limite mínimo é concluída.

### Como funciona o pré-escalonamento

Quando o limite mínimo para ECPUs /segundo ou armazenamento de dados é atualizado por meio do console CLI/API, ou quando esse novo limite está disponível em 1 hora. ElastiCache O Serverless

suporta 30 ECPUs K/segundo em um cache vazio e até 90 ECPUs K/seg ao usar o recurso Ler da Réplica. ElastiCache pode ECPUs dobrar/segundo a cada 10-12 minutos. Essa velocidade de escalabilidade é suficiente para a maioria das cargas de trabalho. Se você prevê que um próximo evento de escalabilidade possa exceder essa taxa, recomendamos definir o mínimo de ECPUs / segundo para o ECPUs pico/seg 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. ECPUs ou o uso de armazenamento de dados que exceda o limite mínimo de uso será cobrado pela taxa normal. Por exemplo, se você definir um limite mínimo de uso de 100.000 por ECPUs segundo, será cobrado pelo menos 1,224 USD por hora (usando preços 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 ECPUs 30K/segundo por slot (ECPUs90K/segundo ao usar Read from Replica usando conexões). READONLY Como prática recomendada, seu aplicativo deve garantir que a distribuição de chaves nos OSS slots Valkey ou Redis e o tráfego entre as 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 AWS Management Console 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 Ação e Modificar.
6. Em Limites de uso, defina os limites apropriados de memória ou computação.
7. Clique em Previsualizar alterações e em Salvar alterações.

## Definindo limites de escala usando o AWS CLI

Para alterar os limites de CLI escala usando o. modify-serverless-cache API

### Linux

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

### Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

## Removendo limites de escala usando o CLI

Para remover os limites de escala usando oCLI, 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 autoprojeados

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 usa. ElastiCache permite que você escale seu cache para atender à demanda.

### Note

Se um OSS cluster Valkey ou Redis for replicado em uma ou mais regiões, essas regiões serão escaladas em ordem. Ao aumentar a escala, as regiões secundárias são escaladas primeiro e depois a região primária. Ao reduzir a escala, a região primária é a primeira e, em seguida, todas as regiões secundárias.

Ao atualizar a versão do mecanismo, o pedido é Região secundária e, em seguida, Região primária.

### Tópicos

- [Clusters de escalabilidade para Memcached](#)
- [Dimensionamento de clusters para Valkey ou Redis OSS \(modo de cluster desativado\)](#)
- [Dimensionamento de nós de réplica para Valkey ou Redis OSS \(modo de cluster desativado\)](#)
- [Dimensionamento de clusters em Valkey ou Redis OSS \(modo de cluster ativado\)](#)

## Clusters de escalabilidade para Memcached

As tabelas a seguir podem ajudá-lo a encontrar o tópico correto para as ações de escalabilidade que você deseja realizar.

### Dimensionando clusters do Memcached

Os clusters do Memcached são compostos de 1 a 60 nós. Escalar um cluster Memcached é tão fácil quanto adicionar ou remover nós do cluster.

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.

## Escalar clusters Memcached

Ação	Tópico
Aumento de escala	<a href="#">Adição de nós a um cluster</a>
Redução de escala	<a href="#">Exclusão de nós de um cluster</a>
Alteração nos tipos de nó	<a href="#">Escalabilidade vertical do Memcached</a>

Os clusters do Memcached são compostos de 1 a 60 nós. Escalar um cluster Memcached é tão fácil quanto adicionar ou remover nós do cluster.

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.

## Tópicos

- [Escalar o Memcached horizontalmente](#)
- [Escalabilidade vertical do Memcached](#)

## Escalar o Memcached horizontalmente

O mecanismo Memcached oferece suporte para particionar seus dados em vários nós. Por isso, os clusters Memcached escalam horizontalmente com facilidade. Um cluster Memcached pode ter de 1 a 60 nós. 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 [Identifique automaticamente os nós em 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.

### Escalabilidade vertical do Memcached

Quando você expande ou reduz seu cluster Memcached, 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 [Select cache node size](#).

### Tópicos

- [Escalabilidade vertical do Memcached \(console\)](#)
- [Escalabilidade vertical do Memcached \(AWS CLI\)](#)
- [Escalabilidade vertical do Memcached \(ElastiCache API\)](#)

### Escalabilidade vertical do Memcached (console)

O procedimento a seguir mostra como escalar seu cluster verticalmente usando o ElastiCache console.

#### Para escalar um cluster do Memcached verticalmente (console)

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 [Encontrando os endpoints de um cluster \(console\) \(Memcached\)](#).
3. Exclua o cluster antigo. Para obter mais informações, consulte [Deleting a new node in Memcached](#).

### Escalabilidade vertical do Memcached (AWS CLI)

O procedimento a seguir demonstra passo a passo como escalonar seu cluster de cache Memcached verticalmente usando a AWS CLI.

Para escalonar um cluster de cache Memcached verticalmente (AWS CLI)

1. Crie um novo cluster de cache com o novo tipo de nó. Para obter mais informações, consulte [Criação de clusters com 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 de cache antigo. Para obter mais informações, consulte [Usando o AWS CLI para excluir um ElastiCache cluster](#).

### Escalabilidade vertical do Memcached (ElastiCache API)

O procedimento a seguir mostra como escalar verticalmente seu cluster de cache do Memcached usando o ElastiCache API

Para escalar verticalmente um cluster de cache do Memcached () ElastiCache API

1. Crie um novo cluster de cache com o novo tipo de nó. Para ter mais informações, consulte [Criação de um cluster para Memcached \(\) ElastiCache API](#)
2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster de cache. Para obter mais informações, consulte [Encontrando endpoints \(\) ElastiCache API](#).
3. Exclua o cluster de cache antigo. Para obter mais informações, consulte [Usando o ElastiCache API](#).



## Dimensionamento de clusters para Valkey ou Redis OSS (modo de cluster desativado)

Os clusters Valkey ou Redis OSS (modo de cluster desativado) podem ser clusters de um único nó com 0 fragmentos ou clusters de vários nós com 1 fragmento. Os clusters de nó único usam esse único nó para leituras e gravações. Os clusters de vários nós têm sempre 1 nó como o nó primário principal de leitura/gravação, e 0 a 5 nós de réplica somente de leitura.

### Tópicos

- [Dimensionamento de clusters de nó único para Valkey ou Redis OSS \(modo de cluster desativado\)](#)

### Dimensionamento de clusters Valkey ou Redis OSS

Ação	Valkey ou Redis OSS (modo de cluster desativado)	Valkey ou Redis OSS (modo de cluster ativado)
Redução de escala	<a href="#">Removendo nós de um ElastiCache cluster</a>	<a href="#">Dimensionamento de clusters em Valkey ou Redis OSS (modo de cluster ativado)</a>
Aumento de escala	<a href="#">Adição de nós a um cluster</a>	<a href="#">Refragmentação online para Valkey ou Redis OSS (modo de cluster ativado)</a>
Alteração nos tipos de nó	<p>Para um tipo de nó maior:</p> <ul style="list-style-type: none"> <li>• <a href="#">Escalando clusters Valkey ou Redis de nó único OSS</a></li> <li>• <a href="#">Ampliando clusters Valkey ou Redis OSS com réplicas</a></li> </ul> <p>Para um tipo de nó menor:</p> <ul style="list-style-type: none"> <li>• <a href="#">Reduzindo clusters Valkey ou Redis de nó único OSS</a></li> <li>• </li> </ul>	<a href="#">Escalabilidade vertical online com modificação do tipo de nó</a>

Ação	Valkey ou Redis OSS (modo de cluster desativado)	Valkey ou Redis OSS (modo de cluster ativado)
	<a href="#"><u>Reduzindo clusters Valkey ou Redis OSS com réplicas</u></a>	
Alteração do número de grupos de nós	Não é compatível com clusters Valkey ou Redis OSS (modo de cluster desativado)	<a href="#"><u>Dimensionamento de clusters em Valkey ou Redis OSS (modo de cluster ativado)</u></a>

## Sumário

- [Dimensionamento de clusters de nó único para Valkey ou Redis OSS \(modo de cluster desativado\)](#)
  - [Escalando clusters Valkey ou Redis de nó único OSS](#)
    - [Escalando clusters de nó único para Valkey ou Redis OSS \(modo de cluster desativado\) \(console\)](#)
    - [Escalando clusters de cache Valkey ou Redis OSS de nó único \(\)AWS CLI](#)
    - [Escalando clusters de cache Valkey ou Redis OSS de nó único \(\) ElastiCache API](#)
  - [Reduzindo clusters Valkey ou Redis de nó único OSS](#)
    - [Reduzindo a escala de um cluster Valkey ou Redis OSS de nó único \(console\)](#)
    - [Reduzindo os clusters de cache Valkey ou Redis OSS de nó único \(\)AWS CLI](#)
    - [Reduzindo os clusters de cache Valkey ou Redis OSS de nó único \(\) ElastiCache API](#)

## Dimensionamento de clusters de nó único para Valkey ou Redis OSS (modo de cluster desativado)

Os nós Valkey ou Redis OSS (modo de cluster desativado) devem ser grandes o suficiente para conter todos os dados do cache, além da sobrecarga do Valkey ou do Redis. OSS Para alterar a capacidade de dados do seu cluster Valkey ou Redis OSS (modo de cluster desativado), você deve escalar verticalmente; escalar para um tipo de nó maior para aumentar a capacidade de dados ou reduzir para um tipo de nó menor para reduzir 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 RedisOSS. Para clusters Valkey ou Redis OSS (modo de cluster desativado), recomendamos que seja disponibilizada memória suficiente para Valkey ou Redis. OSS

Você não pode particionar seus dados em vários clusters Valkey ou Redis OSS (modo de cluster desativado). No entanto, se você precisar apenas aumentar ou diminuir a capacidade de leitura do cluster, poderá criar um cluster Valkey ou Redis OSS (modo de cluster desativado) com nós de réplica e adicionar ou remover réplicas de leitura. Para criar um cluster Valkey ou Redis OSS (modo de cluster desativado) com nós de réplica usando seu cluster de OSS cache Valkey ou Redis de nó único como cluster primário, consulte [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)

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 obter mais informações, consulte [Aumento da capacidade de leitura](#) ou [Redução da capacidade de leitura](#).

Além de poderem escalar a capacidade de leitura, os clusters Valkey ou Redis OSS (modo de cluster desativado) com réplicas oferecem outras vantagens comerciais. Para obter mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

#### Important

Se seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, antes de começar a escalar, certifique-se de ter um grupo de parâmetros personalizado que reserve a quantidade correta de memória para 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 [Gerenciando a memória reservada para Valkey e Redis OSS](#).

## Tópicos

- [Escalando clusters Valkey ou Redis de nó único OSS](#)
- [Reduzindo clusters Valkey ou Redis de nó único OSS](#)

## Escalando clusters Valkey ou Redis de nó único OSS

Ao escalar um OSS cluster Valkey ou Redis de nó único, ElastiCache executa o seguinte processo, independentemente de você usar o ElastiCache console AWS CLI, o ou o ElastiCache API

1. Um novo cluster de cache com o novo tipo de nó é ativado na mesma zona de disponibilidade que o cluster de cache existente.
2. Os dados do cache no cluster de cache existente são copiados para o novo cluster de cache. A duração desse processo depende do tipo de nó e da quantidade de dados no cluster de cache.
3. As leituras e as gravações são atendidas com o novo cluster de cache. Como os endpoints do novo cluster de cache são os mesmos do que para o cluster de cache antigo, você não precisa atualizar os endpoints no seu aplicativo. Você notará uma breve interrupção (alguns segundos) das leituras e gravações do nó primário enquanto a DNS entrada é atualizada.
4. ElastiCache exclui o antigo cluster de cache. 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 aumento de OSS escala do Valkey ou do Redis será bloqueada se você tiver uma atualização do motor programada para a próxima janela de manutenção. Para obter mais informações sobre janelas de manutenção, consulte [Gerenciando a manutenção ElastiCache 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

Operações pendentes	Operações bloqueadas
	Atualização imediata do mecanismo

Se você tiver uma operação pendente que está bloqueando você, será possível tomar uma destas providências.

- Agende sua operação de OSS expansão do Valkey ou do Redis para a próxima janela de manutenção desmarcando a caixa de seleção Aplicar imediatamente (CLI use: `--no-apply-immediately`). API `ApplyImmediately=false`
- Espere até a próxima janela de manutenção (ou depois) para realizar sua operação de OSS expansão do Valkey ou do Redis.
- Adicione a atualização do OSS mecanismo Valkey ou Redis a essa modificação do cluster de cache com a caixa de seleção Aplicar imediatamente selecionada (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 de nó único OSS (modo de cluster desativado) usando o ElastiCache console, o AWS CLI ou. ElastiCache API

#### Important

Se seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, antes de começar a escalar, certifique-se de ter um grupo de parâmetros personalizado que reserve a quantidade correta de memória para 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 [Gerenciando a memória reservada para Valkey e Redis OSS](#).

## Escalando clusters de nó único para Valkey ou Redis OSS (modo de cluster desativado) (console)

O procedimento a seguir descreve como escalar um OSS cluster Valkey ou Redis de nó único usando o ElastiCache Management Console. Durante esse processo, seu OSS cluster Valkey ou Redis continuará atendendo às solicitações com o mínimo de tempo de inatividade.

Para escalar um OSS cluster Valkey ou Redis de nó único (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou Redis OSS.
3. Na lista de clusters, escolha o cluster que você deseja escalar (ele deve estar executando o mecanismo Valkey ou Redis, não o OSS mecanismo Valkey ou Redis em cluster). 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.

## Escalando clusters de cache Valkey ou Redis OSS de nó único (AWS CLI)

O procedimento a seguir descreve como escalar um cluster de OSS cache Valkey ou Redis de nó único usando o AWS CLI. Durante esse processo, seu OSS cluster Valkey ou Redis continuará atendendo às solicitações com o mínimo de tempo de inatividade.

## Para escalar um cluster de OSS cache Valkey ou Redis de nó único (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 mais ou menos assim (JSONformato).

```
{  
  "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-modificações](#) na Referência da AWS CLI .

2. Modifique seu cluster de cache existente especificando o cluster de cache 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 de cache que você está expandindo.
  - `--cache-node-type`: o novo tipo de nó para o qual você deseja escalar o cluster de cache. 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 mais ou menos assim (JSONformato).

```
{
  "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 Referência da AWS CLI .

3. Se você usou o `--apply-immediately`, verifique o status do novo cluster de cache 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 de cache maior.
  - `--cache-cache cluster-id`— O nome do seu cluster de cache Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster de cache específico em vez de todos os clusters de cache.

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

Para obter mais informações, consulte [describe-cache-clusters](#) na Referência da AWS CLI .

## Escalando clusters de cache Valkey ou Redis OSS de nó único () ElastiCache API

O procedimento a seguir descreve como escalar um cluster de OSS cache Valkey ou Redis de nó único usando o. ElastiCache API Durante esse processo, seu OSS cluster Valkey ou Redis continuará atendendo às solicitações com o mínimo de tempo de inatividade.

Para escalar um cluster de OSS cache Valkey ou Redis de nó único () ElastiCache API

1. Determine os tipos de nós para os quais você pode escalar executando a ElastiCache API `ListAllowedNodeTypeModifications` ação com o parâmetro a seguir.
  - `CacheClusterId`— O nome do cluster de OSS cache Valkey ou Redis de nó único que você deseja ampliar.

```
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](#) na Amazon ElastiCache API Reference.

2. Modifique seu cluster de cache existente especificando o cluster de cache a ser expandido e o novo tipo de nó maior, usando a `ModifyCacheCluster` ElastiCache API ação e os parâmetros a seguir.
  - `CacheClusterId` - O nome do cluster de cache que você está expandindo.

- **CacheNodeType**: o novo tipo de nó maior para o qual você deseja aumentar a escala vertical do cluster de cache. Esse valor deve ser um dos tipos de nós retornados pela `ListAllowedNodeTypeModifications` ação 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-xl  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyCacheCluster](#) na Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, verifique o status do novo cluster de cache usando a ElastiCache API `DescribeCacheClusters` ação com o parâmetro a seguir. Quando o status mudar para `available`, você poderá começar a usar o novo cluster de cache maior.
- **CacheClusterId**— O nome do seu cluster de cache Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster de cache específico em vez de todos os clusters de cache.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=MyRedisCacheCluster
```

```
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeCacheClusters](#) na Amazon ElastiCache API Reference.

## Reduzindo clusters Valkey ou Redis de nó único OSS

As seções a seguir explicam como escalar um OSS cluster Valkey ou Redis de nó único para um tipo de nó menor. Garantir que o novo tipo de nó menor seja grande o suficiente para acomodar todos os dados e a OSS sobrecarga do Valkey ou do Redis é importante para o sucesso a longo prazo do seu novo cluster Valkey ou Redis. Para obter mais informações, consulte [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do 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

- [Reduzindo a escala de um cluster Valkey ou Redis OSS de nó único \(console\)](#)
- [Reduzindo os clusters de cache Valkey ou Redis OSS de nó único \(AWS CLI\)](#)
- [Reduzindo os clusters de cache Valkey ou Redis OSS de nó único \(ElastiCache API\)](#)

### Reduzindo a escala de um cluster Valkey ou Redis OSS de nó único (console)

O procedimento a seguir mostra como escalar seu OSS cluster Valkey ou Redis de nó único para um tipo de nó menor usando o console. ElastiCache

### Important

Se seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, antes de começar a escalar, certifique-se de ter um grupo de parâmetros personalizado que reserve a quantidade correta de memória para 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 [Gerenciando a memória reservada para Valkey e Redis OSS](#).

## Para reduzir seu OSS cluster Valkey ou Redis 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 seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, certifique-se de ter um grupo de parâmetros personalizado para reservar a quantidade correta de memória para seu novo tipo de nó.

Alternativamente, você pode modificar seu `parameter group` personalizado para usar `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciando a memória reservada para Valkey e Redis OSS](#).

3. Faça login no AWS Management Console 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 e não o OSS mecanismo Valkey ou Redis em cluster. 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 dimensionado para o novo tipo de nó. Não é necessário atualizar os endpoints no seu aplicativo.

## Reduzindo os clusters de cache Valkey ou Redis OSS de nó único (AWS CLI)

O procedimento a seguir descreve como reduzir um cluster de OSS cache Valkey ou Redis de nó único usando o AWS CLI

## Para reduzir um cluster de OSS cache Valkey ou Redis de nó único ( )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 mais ou menos assim (JSONformato).

```
{  
  "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-modificações](#) na Referência da AWS CLI .

2. Modifique seu cluster de cache existente especificando o cluster de cache 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 de cache que você está reduzindo a escala vertical.
  - `--cache-node-type`: o novo tipo de nó para o qual você deseja escalar o cluster de cache. 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 mais ou menos assim (JSONformato).

```
{
  "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 Referência da AWS CLI .

3. Se você usou o `--apply-immediately`, verifique o status do novo cluster de cache 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 de cache maior.

- `--cache-cache cluster-id`— O nome do seu cluster de cache Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster de cache específico em vez de todos os clusters de cache.

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

Para obter mais informações, consulte [describe-cache-clusters](#) na Referência da AWS CLI .

## Reduzindo os clusters de cache Valkey ou Redis OSS de nó único () ElastiCache API

O procedimento a seguir descreve como aumentar a escala de um cluster de OSS cache Valkey ou Redis de nó único usando o. ElastiCache API

Para reduzir um cluster de OSS cache Valkey ou Redis de nó único () ElastiCache API

1. Determine os tipos de nós para os quais você pode reduzir executando a ElastiCache API `ListAllowedNodeTypeModifications` ação com o parâmetro a seguir.
  - `CacheClusterId`— O nome do cluster de OSS cache Valkey ou Redis de nó único que você deseja reduzir.

```
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](#) na Amazon ElastiCache API Reference.

2. Modifique seu cluster de cache existente especificando o cluster de cache a ser expandido e o novo tipo de nó maior, usando a `ModifyCacheCluster` ElastiCache API ação e os parâmetros a seguir.
  - `CacheClusterId`: o nome do cluster de cache que você está reduzindo a escala vertical.

- **CacheNodeType**: o novo tipo de nó menor para o qual você deseja reduzir a escala vertical do cluster de cache. Esse valor deve ser um dos tipos de nó retornados pela `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, verifique o status do novo cluster de cache usando a ElastiCache API `DescribeCacheClusters` ação com o parâmetro a seguir. Quando o status mudar para `available` (disponível), você poderá começar a usar o novo cluster de cache menor.
  - **CacheClusterId**— O nome do seu cluster de cache Valkey ou Redis OSS de nó único. Use esse parâmetro para descrever um cluster de cache específico em vez de todos os clusters de cache.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=MyRedisCacheCluster
```

```
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [DescribeCacheClusters](#) na Amazon ElastiCache API Reference.

## Dimensionamento de nós de réplica para Valkey ou Redis OSS (modo de cluster desativado)

Um OSS cluster Valkey ou Redis com nós de réplica (chamado de grupo de replicação noAPI/CLI) fornece alta disponibilidade por meio de replicação que tem Multi-AZ com failover automático ativado. Um cluster com nós de réplica é uma coleção lógica de até seis OSS nós Valkey ou Redis em que um nó, o Primário, é capaz de atender às 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 o Redis OSS (modo de cluster desativado) não suportam o particionamento de seus dados em vários clusters, cada nó em um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) contém todo o conjunto de dados do cache. Os clusters Valkey ou Redis OSS (modo de cluster ativado) suportam o particionamento de seus dados em 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 RedisOSS. Para OSS clusters Valkey ou Redis com réplicas, recomendamos que seja disponibilizada memória suficiente para Valkey ou Redis. OSS

### Tópicos

- [Ampliando clusters Valkey ou Redis OSS com réplicas](#)
- [Reduzindo clusters 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 OSS cluster Valkey e Redis desativado versus ativado](#)
- [Minimizando o tempo de inatividade ElastiCache usando o Multi-AZ com Valkey e Redis OSS](#)

- [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#)

## Tópicos

- [Ampliando clusters Valkey ou Redis OSS com réplicas](#)
- [Reduzindo clusters Valkey ou Redis OSS com réplicas](#)
- [Aumento da capacidade de leitura](#)
- [Redução da capacidade de leitura](#)

## Ampliando clusters Valkey ou Redis OSS com réplicas

ElastiCache A Amazon fornece console e API suporte para escalar seu grupo de replicação Valkey ou Redis OSS (modo de cluster desativado). CLI

Quando o processo de expansão 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 DNS entradas para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para Valkey 7.2 e superior ou Redis OSS 5.0.5 e superior, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atende às solicitações recebidas. Na OSS versão 4.0.10 e versões anteriores do Redis, você pode observar uma breve interrupção das leituras e gravações nas versões anteriores do nó primário enquanto a DNS entrada 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 OSS expansão do Valkey ou do Redis será bloqueada se você tiver um upgrade de mecanismo agendado 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.

- Agende sua operação de OSS expansão do Valkey ou do Redis para a próxima janela de manutenção desmarcando a caixa de seleção Aplicar imediatamente (CLI use:, use:--no-apply-immediately). API ApplyImmediately=false
- Espere até a próxima janela de manutenção (ou depois) para realizar sua operação de OSS expansão Valkey ou Redis.
- Adicione a atualização do OSS mecanismo Valkey ou Redis a essa modificação do cluster de cache com a caixa de seleção Aplicar imediatamente selecionada (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 OSS cluster Valkey ou Redis com réplicas usando o ElastiCache console AWS CLI, o e o. ElastiCache API

#### Important

Se seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, antes de começar a escalar, certifique-se de ter um grupo de parâmetros personalizado que reserve a quantidade correta de memória para 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 [Gerenciando a memória reservada para Valkey e Redis OSS](#).

## Escalando um OSS cluster Valkey ou Redis 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, pode haver uma breve interrupção das leituras e gravações de outras versões do nó primário enquanto a DNS entrada é atualizada. Você



pode ver menos de 1 segundo de inatividade para nós executados nas versões 5.0.6 e superiores e alguns segundos para versões mais antigas.

Para escalar o OSS cluster Valkey ou Redis com réplicas (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters Redis OSS
3. Na lista de clusters, escolha o cluster que você deseja expandir. Esse cluster deve estar executando o mecanismo Valkey ou Redis e não o OSS mecanismo Valkey ou Redis em cluster. 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 dimensionado para o novo tipo de nó. Não é necessário atualizar os endpoints no seu aplicativo.

Escalando um grupo de OSS replicação Valkey ou Redis ( )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 DNS entradas para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para Valkey 7.2 e superior ou Redis OSS 5.0.5 e superior, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atende às solicitações recebidas. Na versão 4.0.10 e versões anteriores, você pode notar uma breve interrupção das leituras e gravações nas versões anteriores do nó primário enquanto a DNS entrada é atualizada.

O tempo necessário para expandir até um tipo de nó maior varia, dependendo do tipo de nó e da quantidade de dados no seu cluster de cache atual.

Para escalar um grupo de OSS replicação Valkey ou Redis ( )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
```

A saída dessa operação é semelhante a esta (JSONformato).

```
{  
  "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-modificações](#) na Referência da AWS CLI .

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 de cache nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pelo `list-allowed-node-type-modifications` comando 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
```

A saída desse comando é mais ou menos assim (JSONformato).

```
{
  "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 Referência da AWS CLI .

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 está sendo modificado, você pode ver menos de 1 segundo de inatividade para os nós executados nas versões 5.0.6 e superiores e uma breve interrupção das leituras e gravações das versões mais antigas do nó primário enquanto a DNS entrada 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 OSS replicação Valkey ou Redis () ElastiCache API

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó maior usando o ElastiCache API Para Valkey 7.2 e superior ou Redis OSS 5.0.5 e superior, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atende às solicitações recebidas. Na versão Redis OSS 4.0.10 e versões anteriores, você pode observar uma breve interrupção das leituras e gravações nas versões anteriores do nó primário enquanto a DNS entrada é atualizada.

O tempo necessário para expandir até um tipo de nó maior varia, dependendo do tipo de nó e da quantidade de dados no seu cluster de cache atual.

## Para escalar um grupo de OSS replicação Valkey ou Redis () ElastiCache API

1. Determine quais tipos de nós você pode escalar usando a ElastiCache API `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando a `ModifyRedplicationGroup` ElastiCache API ação e com os parâmetros a seguir.

- `ReplicationGroupId` - o nome do grupo de replicação.
- `CacheNodeType`: o novo tipo de nó maior dos clusters de cache nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a ElastiCache API `DescribeReplicationGroups` ação com os seguintes parâmetros. 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](#) na Amazon ElastiCache API Reference.

## Reduzindo clusters Valkey ou Redis OSS com réplicas

As seções a seguir explicam como escalar um cluster de cache Valkey ou Redis OSS (modo de cluster desativado) com nós de réplica até 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 [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do 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 seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, antes de começar a escalar, certifique-se de ter um grupo de parâmetros personalizado que reserve a quantidade correta de memória para 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 [Gerenciando a memória reservada para Valkey e Redis OSS](#).

## Reduzindo a escala de um OSS cluster Valkey ou Redis com réplicas (console)

O processo a seguir escala seu OSS cluster Valkey ou Redis com nós de réplica para um tipo de nó menor usando o console. ElastiCache

### Para reduzir um OSS cluster Valkey ou Redis 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 seu grupo de parâmetros costuma `reserved-memory` reservar memória para OSS sobrecarga do Valkey ou do Redis, certifique-se de ter um grupo de parâmetros personalizado para reservar a quantidade correta de memória para seu novo tipo de nó.



Alternativamente, você pode modificar seu parameter group personalizado para usar `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciando a memória reservada para Valkey e Redis OSS](#).

3. Faça login no AWS Management Console 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 e não o OSS mecanismo Valkey ou Redis em cluster. 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 dimensionado para o novo tipo de nó. Não é necessário atualizar os endpoints no seu aplicativo.

## Reduzindo um grupo de OSS replicação Valkey ou Redis ()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 DNS entradas para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Valkey 7.2 superior ou o Redis OSS 5.0.5 e superior, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atende às solicitações recebidas. Na versão 4.0.10 e versões anteriores, você pode notar uma breve interrupção das leituras e gravações nas versões anteriores do nó primário enquanto a DNS entrada é atualizada.

No entanto, as leituras dos clusters de cache de réplica de leitura continuam ininterruptas.

O tempo necessário para reduzir até um tipo de nó menor varia, dependendo do tipo de nó e da quantidade de dados no seu cluster de cache atual.

Para reduzir um grupo de OSS replicação Valkey ou Redis ( )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
```

A saída dessa operação é semelhante a esta (JSONformato).

```
{  
  "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-modificações](#) na Referência da AWS CLI .

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 de cache nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pelo `list-allowed-node-type-modifications` comando 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
```

A saída desse comando é mais ou menos assim (JSONformato).

```
{"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 Referência da AWS CLI .

- 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 OSS replicação Valkey ou Redis () ElastiCache API

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando o. ElastiCache API Durante esse processo, ElastiCache atualiza as DNS entradas para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para Valkey 7.2 e superior ou Redis OSS 5.0.5 e superior, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atende às solicitações recebidas. Na OSS versão 4.0.10 e versões anteriores do Redis, você pode notar uma breve

interrupção das leituras e gravações nas versões anteriores do nó primário enquanto a DNS entrada é atualizada. No entanto, as leituras dos clusters de cache de réplica de leitura continuam ininterruptas.

O tempo necessário para reduzir até um tipo de nó menor varia, dependendo do tipo de nó e da quantidade de dados no seu cluster de cache atual.

Para reduzir um grupo de OSS replicação Valkey ou Redis () ElastiCache API

1. Determine quais tipos de nós você pode reduzir usando a ElastiCache API `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando a `ModifyRedplicationGroup` ElastiCache API ação e com os parâmetros a seguir.

- `ReplicationGroupId` - o nome do grupo de replicação.
- `CacheNodeType`: o novo tipo de nó menor dos clusters de cache nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a ElastiCache API `DescribeReplicationGroups` ação com os seguintes parâmetros. 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](#) na 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 RedisOSS.

Você pode escalar a capacidade de leitura do OSS cluster Valkey ou Redis usando o ElastiCache console AWS CLI, o ou o. ElastiCache API Para obter mais informações, consulte [Adicionar uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#).

## Redução da capacidade de leitura

Para diminuir a capacidade de leitura, exclua uma ou mais réplicas de leitura do seu OSS cluster Valkey ou Redis com réplicas (chamado de grupo de replicação no/). 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 [Excluindo uma réplica de leitura para Valkey ou Redis OSS \(modo de cluster desativado\)](#).

## Dimensionamento de clusters em Valkey ou Redis OSS (modo de cluster ativado)

Conforme a demanda em seus clusters muda, você pode decidir melhorar o desempenho ou reduzir custos alterando o número de fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado). 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`, `BytesUseForCachee`.

- **CPU ou gargalo de rede:**

Se os problemas de latência/throughput estão enfraquecendo seu cluster, pode ser necessário aumentar a escala 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: `FreeableMemorySwapUsageBytesUseForCacheCPUUtilization`, `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. A quantidade de degradação que você enfrenta depende de sua CPU utilização normal e de seus dados.

Há duas maneiras de escalar seu cluster Valkey ou Redis OSS (modo de cluster ativado): 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 reestilhaçamento online permite expandir/reduzir enquanto o cluster continua veiculando solicitações de entrada.

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 processo de escalabilidade vertical online permite expandir/reduzir enquanto o cluster continua veiculando solicitações de entrada.

Se você estiver reduzindo o tamanho e a capacidade de memória do cluster, aumentando ou diminuindo a escala, certifique-se de que a nova configuração tenha memória suficiente para seus dados e a sobrecarga do Valkey ou do Redis. OSS

Para obter mais informações, consulte [Select cache node size](#).

## Sumário

- [Refragmentação offline para Valkey ou Redis OSS \(modo de cluster ativado\)](#)
- [Refragmentação online para Valkey ou Redis OSS \(modo de cluster ativado\)](#)
  - [Adição de fragmentos com refragmentação online](#)
  - [Remoção de fragmentos com refragmentação online](#)
    - [Remoção de fragmentos \(console\)](#)
    - [Remoção de fragmentos \(AWS CLI\)](#)
    - [Removendo fragmentos \(\) ElastiCache API](#)
  - [Rebalanceamento de frgamentos online](#)
    - [Rebalanceamento de fragmentos online \(Console\)](#)
    - [Rebalanceamento de fragmentos online \(AWS CLI\)](#)
    - [Rebalanceamento de fragmentos online \(ElastiCache API\)](#)
- [Escalabilidade vertical online com modificação do tipo de nó](#)

- [Aumento de escala vertical online](#)
  - [Ampliando clusters de OSS cache Valkey ou Redis \(console\)](#)
  - [Aumentando a escala de clusters de OSS cache Valkey ou Redis \(\)AWS CLI](#)
  - [Aumentando a escala de clusters de OSS cache Valkey ou Redis \(\) ElastiCache API](#)
- [Redução de escala vertical online](#)
  - [Reduzindo os clusters de OSS cache Valkey ou Redis \(console\)](#)
  - [Reduzindo os clusters de OSS cache Valkey ou Redis \(\)AWS CLI](#)
  - [Reduzindo os clusters de OSS cache Valkey ou Redis \(\) ElastiCache API](#)

### Refragmentação offline para Valkey ou Redis OSS (modo de cluster ativado)

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. Ao refragmentar e reequilibrar off-line, além de alterar o número de fragmentos em seu grupo de replicação, você pode fazer o seguinte:

#### Note

A refragmentação offline não é suportada em OSS clusters Valkey ou Redis com a classificação por níveis de dados ativada. Para obter mais informações, consulte [Hierarquização de dados em 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 Valkey ou Redis OSS (modo de cluster ativado), agrupe offline

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 [Encontrando pontos de extremidade de conexão em ElastiCache](#).

Refragmentação online para Valkey ou Redis OSS (modo de cluster ativado)

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 (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:

- **Escale** — aumente a capacidade de leitura e gravação adicionando fragmentos (grupos de nós) ao seu cluster Valkey ou Redis OSS (modo de cluster ativado) (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.

- **Amplie** — reduza a capacidade de leitura e gravação e, portanto, os custos, removendo fragmentos do seu cluster Valkey ou Redis OSS (habilitado para o modo de cluster).
- **Rebalancear** — Mova os espaços de chave entre os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) para que sejam distribuídos o mais igualmente possível 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 e mais recente ou a versão 3.2.10 ou mais recente OSS do motor Redis. 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

ElastiCache Com o Valkey 7.2 e superior e com o Redis OSS versão 7 e 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 seu aplicativo usa [Funções](#), recomendamos carregar todas as suas funções em todos os fragmentos antes de escalar 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 o AWS Management Console, o e o. AWS CLI ElastiCache API

### Adição de fragmentos com refragmentação online

Você pode adicionar fragmentos ao seu cluster Valkey ou Redis OSS (modo de cluster ativado) usando o AWS Management Console,, AWS CLI ou. ElastiCache API Quando você adiciona fragmentos a um cluster Valkey ou Redis OSS (modo de cluster ativado), todas as tags nos fragmentos existentes são copiadas para os novos fragmentos.

### Adição de fragmentos (console)

Você pode usar o AWS Management Console para adicionar um ou mais fragmentos ao seu cluster Valkey ou Redis OSS (modo de cluster ativado). O procedimento a seguir descreve o processo.

Para adicionar fragmentos ao seu cluster Valkey ou Redis OSS (modo de cluster ativado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters Redis OSS.
3. Localize e escolha o nome, e não a caixa à esquerda do nome do cluster, do cluster Valkey ou Redis OSS (modo de cluster ativado) ao qual você deseja adicionar fragmentos.

#### Tip

Valkey ou Redis OSS (modo de cluster ativado) mostra Clustered Valkey ou Clustered Redis na coluna Modo OSS

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 em seu cluster Valkey ou Redis OSS (modo de cluster ativado) adicionando fragmentos usando o 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`: optional. 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 Valkey ou Redis OSS (modo de cluster ativado) chamado `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.

### Adição de fragmentos (ElastiCache API)

Você pode usar o 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`: optional. 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 o ElastiCache API

### Example Adicionar fragmentos

O exemplo a seguir adiciona grupos de nós ao cluster Valkey ou Redis OSS (modo de cluster ativado) `my-cluster`, portanto, há um total de quatro grupos de nós quando a operação é 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.AvailabilityZone
east-2c
  &Version=2015-02-02
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20171002T192317Z
  &X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyReplicationGroupShardConfiguration](#) na ElastiCache API Referência.

## Remoção de fragmentos com refragmentação online

Você pode remover fragmentos do seu cluster Valkey ou Redis OSS (modo de cluster ativado) usando o AWS Management Console,, AWS CLI ou. ElastiCache API

### Tópicos

- [Remoção de fragmentos \(console\)](#)
- [Remoção de fragmentos \(AWS CLI\)](#)
- [Removendo fragmentos \(\) ElastiCache API](#)

## Remoção de fragmentos (console)

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) removendo fragmentos usando o. AWS Management Console

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 AWS Management Console 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 Valkey ou Redis OSS (modo de cluster ativado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters Redis OSS.
3. Localize e escolha o nome, e não a caixa à esquerda do nome do cluster, do cluster Valkey ou Redis OSS (modo de cluster ativado) do qual você deseja remover fragmentos.


 Tip

Os clusters Valkey ou Redis OSS (modo de cluster ativado) têm um valor de 1 ou maior na coluna Shards.

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 em seu cluster Valkey ou Redis OSS (modo de cluster ativado) removendo fragmentos usando o 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 Valkey ou Redis OSS (modo de cluster ativado) `my-cluster`, portanto, há um total de dois grupos de nós quando a operação é 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 () ElastiCache API

Você pode usar o 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 o. ElastiCache API

#### 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 o 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 Remover fragmentos

O exemplo a seguir remove dois grupos de nós do cluster Valkey ou Redis OSS (modo de cluster ativado) `my-cluster`, portanto, há um total de dois grupos de nós quando a operação é 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 o AWS Management Console, ou. AWS CLI ElastiCache API

### Tópicos

- [Rebalanceamento de fragmentos online \(Console\)](#)
- [Rebalanceamento de fragmentos online \(AWS CLI\)](#)
- [Rebalanceamento de fragmentos online \(ElastiCache API\)](#)

### Rebalanceamento de fragmentos online (Console)

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado) rebalanceando os fragmentos usando o. AWS Management Console



Para reequilibrar os espaços de chave entre os fragmentos em seu cluster Valkey ou Redis OSS (modo de cluster ativado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters Redis OSS.
3. Escolha o nome, não a caixa à esquerda do nome, do cluster Valkey ou Redis OSS (modo de cluster ativado) que você deseja reequilibrar.

 Tip

Os clusters Valkey ou Redis OSS (modo de cluster ativado) têm um valor de 1 ou maior na coluna Shards.

4. Escolha Rebalance (Rebalancear).
5. Quando solicitado, escolha Rebalance (Rebalancear). Talvez você veja 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 você fizer isso, escolha 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 em seu cluster Valkey ou Redis OSS (modo de cluster ativado) rebalanceando os fragmentos usando o AWS CLI

## Example - Rebalanceando os fragmentos em um cluster

O exemplo a seguir reequilibra os slots no cluster Valkey ou Redis OSS (modo de cluster ativado) `my-cluster` para que os slots sejam distribuídos da forma mais igual 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 online (ElastiCache API)

Você pode usar o 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 o ElastiCache API

## Example - Rebalanceamento do cluster

O exemplo a seguir reequilibra os slots no cluster Valkey ou Redis OSS (modo de cluster ativado) `my-cluster` para que os slots sejam distribuídos da forma mais igual 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 o escalonamento vertical on-line com o Valkey versão 7.2 ou nova, ou o Redis OSS versão 3.2.10 ou mais recente, você pode escalar seus clusters Valkey ou Redis dinamicamente com o mínimo de tempo de inatividade. Isso permite que seu OSS cluster Valkey ou Redis atenda às solicitações mesmo durante a escalabilidade.

### 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 [Hierarquização de dados em ElastiCache](#).

Você pode fazer o seguinte:

- Aumente a capacidade de leitura e gravação ajustando o tipo de nó do seu OSS cluster Valkey ou Redis 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 fluxo suave de expansão/redução, faça o seguinte:

- Certifique-se de ter capacidade suficiente ENI (Elastic Network Interface). 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 [Gerenciando a 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](#)
  - [Ampliando clusters de OSS cache Valkey ou Redis \(console\)](#)
  - [Aumentando a escala de clusters de OSS cache Valkey ou Redis \(AWS CLI\)](#)
  - [Aumentando a escala de clusters de OSS cache Valkey ou Redis \(ElastiCache API\)](#)
- [Redução de escala vertical online](#)
  - [Reduzindo os clusters de OSS cache Valkey ou Redis \(console\)](#)
  - [Reduzindo os clusters de OSS cache Valkey ou Redis \(AWS CLI\)](#)
  - [Reduzindo os clusters de OSS cache Valkey ou Redis \(ElastiCache API\)](#)

## Aumento de escala vertical online

### Tópicos

- [Ampliando clusters de OSS cache Valkey ou Redis \(console\)](#)
- [Aumentando a escala de clusters de OSS cache Valkey ou Redis \(\)AWS CLI](#)
- [Aumentando a escala de clusters de OSS cache Valkey ou Redis \(\) ElastiCache API](#)

### Ampliando clusters de OSS cache Valkey ou Redis (console)

O procedimento a seguir descreve como escalar um OSS cluster Valkey ou Redis usando o ElastiCache Management Console. Durante esse processo, seu cluster continuará atendendo às solicitações com o mínimo de tempo de inatividade.

### Para escalar um OSS cluster Valkey ou Redis (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters 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.

## Aumentando a escala de clusters de OSS cache Valkey ou Redis ( )AWS CLI

O procedimento a seguir descreve como escalar um cluster de OSS cache Valkey ou Redis usando o AWS CLI. Durante esse processo, seu cluster continuará atendendo às solicitações com o mínimo de tempo de inatividade.

Para escalar um cluster de OSS cache Valkey ou Redis ( )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 mais ou menos assim (JSONformato).

```
{  
  "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-modificações](#) na Referência da AWS CLI .

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 de cache. 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 mais ou menos assim (JSONformato).

```
{
  "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 Referência da AWS CLI .

3. Se você usou o `--apply-immediately`, verifique o status do cluster de cache 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 de cache maior.

## Aumentando a escala de clusters de OSS cache Valkey ou Redis () ElastiCache API

O processo a seguir dimensiona seu cluster de cache do tipo de nó atual para um novo tipo de nó maior usando ElastiCache API o. Durante esse processo, ElastiCache atualiza as DNS entradas 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 versões posteriores do Redis OSS 5.0.5 e versões posteriores, você pode escalar clusters habilitados para failover automático enquanto o cluster continua on-line e atende às solicitações recebidas. Na versão Redis OSS 4.0.10 e versões anteriores, você pode notar uma breve interrupção das leituras e gravações nas versões anteriores do nó primário enquanto a DNS entrada é atualizada.

O tempo necessário para expandir até um tipo de nó maior varia, dependendo do tipo de nó e da quantidade de dados no seu cluster de cache atual.

## Para escalar um cluster de OSS cache Valkey ou Redis () ElastiCache API

1. Determine quais tipos de nós você pode escalar usando a ElastiCache API `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

2. Dimensione seu grupo de replicação atual até o novo tipo de nó usando a `ModifyReplicationGroup` ElastiCache API ação e com os parâmetros a seguir.
  - `ReplicationGroupId` - o nome do grupo de replicação.
  - `CacheNodeType`: o novo tipo de nó maior dos clusters de cache nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a ElastiCache API `DescribeReplicationGroups` ação com os seguintes parâmetros. 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](#) na Amazon ElastiCache API Reference.

## Redução de escala vertical online

### Tópicos

- [Reduzindo os clusters de OSS cache Valkey ou Redis \(console\)](#)
- [Reduzindo os clusters de OSS cache Valkey ou Redis \(AWS CLI\)](#)
- [Reduzindo os clusters de OSS cache Valkey ou Redis \(ElastiCache API\)](#)

### Reduzindo os clusters de OSS cache Valkey ou Redis (console)

O procedimento a seguir descreve como reduzir um OSS cluster Valkey ou Redis usando o ElastiCache Management Console. Durante esse processo, seu OSS cluster Valkey ou Redis continuará atendendo às solicitações com o mínimo de tempo de inatividade.

Para reduzir a escala de um OSS cluster Valkey ou Redis (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters 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.

Reduzindo os clusters de OSS cache Valkey ou Redis (AWS CLI)

O procedimento a seguir descreve como reduzir um cluster de OSS cache Valkey ou Redis usando o AWS CLI. Durante esse processo, seu cluster continuará atendendo às solicitações com o mínimo de tempo de inatividade.

Para reduzir um cluster de OSS cache Valkey ou Redis (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 mais ou menos assim (JSONformato).

```
{  
  "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-modificações](#) na Referência da AWS CLI .

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 de cache. 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 mais ou menos assim (JSONformato).

```
{  
  "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 Referência da AWS CLI .

3. Se você usou `o--apply-immediately`, verifique o status do cluster de cache 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 de cache menor.

## Reduzindo os clusters de OSS cache Valkey ou Redis () ElastiCache API

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando o ElastiCache API. Durante esse processo, seu OSS cluster Valkey ou Redis continuará atendendo às solicitações com o mínimo de tempo de inatividade.

O tempo necessário para reduzir até um tipo de nó menor varia, dependendo do tipo de nó e da quantidade de dados no seu cluster de cache atual.

### Reduzindo a escala () ElastiCache API

#### 1. Determine quais tipos de nós você pode reduzir usando a ElastiCache API

`ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

#### 2. Reduza seu grupo de replicação atual para o novo tipo de nó usando a

`ModifyReplicationGroup` ElastiCache API ação e com os parâmetros a seguir.

- `ReplicationGroupId` - o nome do grupo de replicação.
- `CacheNodeType`: o novo tipo de nó menor dos clusters de cache nesse grupo de replicação. Esse valor deve ser um dos tipos de instância retornados pela `ListAllowedNodeTypeModifications` ação 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](#) na Amazon ElastiCache API Reference.

## Introdução ao JSON Valkey e ao Redis OSS

ElastiCache suporta o formato nativo JavaScript Object Notation (JSON), que é uma forma simples e sem esquemas de codificar conjuntos de dados complexos dentro dos clusters Valkey e Redis. OSS Você pode armazenar e acessar dados de forma nativa usando o formato JavaScript Object Notation (JSON) dentro dos clusters e atualizar os JSON dados armazenados nesses clusters, sem precisar gerenciar código personalizado para serializá-lo e desserializá-lo.

Além de usar as OSS API operações Valkey e Redis para aplicativos que operam em excessoJSON, agora você pode recuperar e atualizar com eficiência partes específicas de um JSON documento sem precisar manipular o objeto inteiro. Isso pode melhorar a performance e reduzir os custos.

Você também pode pesquisar o conteúdo do JSON documento usando a consulta no estilo [GoessnerJSONPath](#).

Depois de criar um cluster com uma versão de mecanismo compatível, o tipo de JSON dados e os comandos associados ficam automaticamente disponíveis. API compatível e RDB compatível com a versão 2 do JSON módulo, para que você possa migrar facilmente os aplicativos Valkey e Redis OSS existentes JSON baseados em. ElastiCache Para obter mais informações sobre os comandos compatíveis, consulte [Comandos Valkey e Redis OSS suportados](#).

As métricas JSON `JsonBasedCmds` relacionadas `JsonBasedCmdsLatency` são incorporadas CloudWatch para monitorar o uso desse tipo de dados. Para obter mais informações, consulte [Métricas para Valkey e Redis OSS](#).

#### Note

Para usar JSON, você deve estar executando o Valkey 7.2 e posterior ou a versão 6.2.6 ou posterior OSS do mecanismo Redis.

## Tópicos

- [JSONvisão geral do tipo de dados](#)
- [Comandos Valkey e Redis OSS suportados](#)

## JSONvisão geral do tipo de dados

ElastiCache suporta vários OSS comandos Valkey e Redis para trabalhar com o tipo de JSON dados. Veja a seguir uma visão geral do tipo de JSON dados e uma lista detalhada dos comandos compatíveis.

## Terminologia

Prazo	Descrição
JSONdocumento	Refere-se ao valor de uma JSON chave.
JSONvalor	Refere-se a um subconjunto de um JSON documento, incluindo a raiz que represent

Prazo	Descrição
	a o documento inteiro. Um valor poderia ser um contêiner ou uma entrada dentro de um contêiner.
JSONelemento	Equivalente ao JSON valor.

## JSONPadrão suportado

JSONo formato é compatível com os padrões de intercâmbio de [RFCdados 7159](#) e [ECMAJSON-404](#). UTF-8 [Unicode](#) em JSON texto é suportado.

## Elemento raiz

O elemento raiz pode ser de qualquer tipo de JSON dados. Observe que no RFC 4627 anterior, somente objetos ou matrizes eram permitidos como valores raiz. Desde a atualização para RFC 7159, a raiz de um JSON documento pode ser de qualquer tipo de JSON dados.

## Limite de tamanho de documentos

JSONos documentos são armazenados internamente em um formato otimizado para acesso e modificação rápidos. 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 JSON documento é limitado a 64 MB, que é o tamanho da estrutura de dados na memória, não a JSON string. Você pode verificar a quantidade de memória consumida por um JSON documento usando o `JSON.DEBUG MEMORY` comando.

## JSON ACLs

- Semelhante às categorias existentes por tipo de dados (`@string`, `@hash` etc.), uma nova categoria `@json` foi adicionada para simplificar o gerenciamento do acesso a JSON comandos e dados. Nenhum outro OSS comando existente do Valkey ou do Redis é membro da categoria `@json`. Todos os JSON comandos impõem quaisquer restrições e permissões de teclas ou comandos.
- Há cinco OSS ACL categorias existentes do Valkey e do Redis que foram atualizadas para incluir os novos JSON comandos: `@read`, `@write`, `@fast`, `@slow` e `@admin`. A tabela a seguir indica o mapeamento dos JSON comandos para as categorias apropriadas.

## ACL

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

JSONcomando	@read	@write	@fast	@slow	@admin
JSON.NUMMULTBY		y	y		
JSON.OBJECTS	y		y		
JSON.OBJECTLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRINGAPPEND		y	y		
JSON.STRINGLEN	y		y		
JSON.STRINGLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

## Limite de profundidade de aninhamento

Quando um JSON objeto ou matriz tem um elemento que é, em si mesmo, outro JSON objeto ou matriz, diz-se que esse objeto ou matriz interno está “aninhado” 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 Redis JSON oferece suporte a dois tipos de sintaxes de caminho:

- Sintaxe aprimorada — segue a JSONPath sintaxe 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 subscripto 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.
&&	LógicoAND, usado para combinar várias expressões de filtro.
	OR lógico, usado para combinar várias expressões de filtro.

## Exemplos

Os exemplos a seguir são baseados nos XML dados de exemplo [de Goessner](#), que modificamos adicionando 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
  }
}
```



```
}
}
```

Path	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 JSON estrutura.
<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 ISBN número.
<code>\$.book[?(@.price&lt;10)]</code>	Todos os livros mais baratos que US\$ 10.
<code>'\$.book[?(@.price &lt; 10)]'</code>	Todos os livros são mais baratos do que US \$ 10. (O caminho deverá estar entre aspas se contiver espaços em branco.)

Path	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 subscripto de matriz. O índice é baseado em 0.

## Exemplos

Path	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.

## JSONmétricas relacionadas

As seguintes métricas de JSON informações são fornecidas:

Informações	Descrição
json_total_memory_bytes	Memória total alocada aos JSON objetos.
json_num_documents	Número total de documentos no Valkey ou no RedisOSS.

Para consultar as métricas principais, execute o seguinte comando:

```
info json_core_metrics
```

## Como o ElastiCache Valkey e o Redis interagem OSS com JSON

A seção a seguir descreve como o ElastiCache Valkey e o Redis OSS interagem com o JSON tipo de dados.

### 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 caminhos em ElastiCache (RedisOSS) é 128. Por isso, um valor como `$.a.b.c.d...` só pode atingir 128 níveis.

### Processamento de valores numéricos

JSON não tem tipos de dados separados para números inteiros e números de ponto flutuante. Todos eles são chamados de números.

### Representações numéricas:

Quando um JSON número é recebido na entrada, ele é convertido em uma das duas representações binárias internas: um inteiro assinado de 64 bits ou um ponto flutuante de precisão IEEE dupla de 64 bits. A string original e toda a sua formatação não serão retidas. Assim, quando um número é gerado como parte de uma JSON resposta, ele é convertido da representação binária interna em uma string imprimível que usa regras de formatação genéricas. Essas regras podem resultar em uma string diferente da que foi recebida.

### Comandos aritméticos NUMINCRBY e NUMMULTBY:

- Se os dois números forem inteiros e o resultado estiver fora do intervalo de `int64`, ele se tornará automaticamente um número de ponto flutuante de precisão IEEE dupla de 64 bits.
- Se pelo menos um dos números for um ponto flutuante, o resultado será um número de ponto flutuante de precisão IEEE dupla de 64 bits.

- Se o resultado exceder o intervalo de 64 bits IEEE duplos, o comando retornará um OVERFLOW erro.

Para obter uma lista detalhada dos comandos disponíveis, consulte [Comandos Valkey e Redis OSS suportados](#).

### Filtragem direta de matriz

ElastiCache com 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)]`, ElastiCache com Valkey ou Redis OSS retornaria `[1,2,3]` em ambas as circunstâncias.

### Comportamento de indexação de matriz

ElastiCache com Valkey ou Redis OSS permite índices positivos e negativos 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, ElastiCache com o Valkey ou o Redis, OSS 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 inferior, não produziria um resultado.

### Avaliação estrita da sintaxe

O MemoryDB não permite JSON caminhos com sintaxe inválida, mesmo que um subconjunto do caminho contenha um caminho válido. Isso acontece para manter o comportamento correto para nossos clientes.

## Comandos Valkey e Redis OSS suportados

ElastiCache suporta os seguintes comandos Valkey e Redis OSSJSON:

### 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.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** (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **caminho** (obrigatório) — Um JSON caminho.
- **json** (obrigatório) — O JSON valor 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.
- SYNTAXERerro se um dos argumentos json de entrada não for uma JSON string válida.
- 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 última matriz atualizada.
- Erro WRONGTYPE se o valor no caminho não for uma matriz.
- SYNTAXERerro se um dos argumentos json de entrada não for uma JSON string 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

Pesquisa a primeira ocorrência de um JSON valor escalar nas matrizes do 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 (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- caminho (obrigatório) — Um JSON caminho.
- json-scalar (obrigatório) - O valor escalar a ser pesquisado. JSONescalar 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 (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **caminho (obrigatório)** — Um JSON caminho.
- **índice (obrigatório)** -Um índice de matriz antes do qual os valores são inseridos.
- **json (obrigatório)** — O JSON valor 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 (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. 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:

- Matriz de strings em massa. Cada elemento é um nome de chave no objeto.
- 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.
- `WRONGTYPE` 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\\"]]]'
(error) SYNTAXERR Failed to parse JSON string due to syntax error
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 $[*]
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 $[*]
```

- 1) (integer) 0
- 2) (nil)
- 3) (integer) 2
- 4) (integer) 3
- 5) (nil)

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 $[3]
1) (integer) 3

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

## JSON.ARRPOP

Remove e retorna elemento no índice da matriz. Exibir uma matriz vazia retorna nulo.

Sintaxe

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

- **chave (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. 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:

- Cadeia de caracteres em massa, que representa o JSON valor 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.ARRINSERT <key> <path> <start> <end>
```

- chave (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- caminho (obrigatório) — Um JSON caminho.
- 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 OUTFBOUNDARIES 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** (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path** (opcional) — Um JSON caminho. 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.

## Exemplos

```
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] — Relata o uso da memória em bytes de um JSON valor. O caminho assumirá o padrão da raiz se não for fornecido.
- FIELDS<key>[caminho] — Informa o número de campos no caminho do documento especificado. O caminho assumirá o padrão da raiz se não for fornecido. Cada JSON valor que não seja do contêiner conta como um campo. Objetos e matrizes contam recursivamente um campo para cada um dos valores JSON contidos. Cada valor de contêiner, exceto o contêiner raiz, conta como um campo adicional.
- HELP— Imprime mensagens de ajuda do comando.

## Sintaxe

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

Depende do subcomando:

## MEMORY

- Se o caminho for uma sintaxe aprimorada:
  - Retorna uma matriz de números inteiros que representam o tamanho da memória (em bytes) do JSON valor em cada caminho.
  - Retorna uma matriz vazia se a chave Valkey ou Redis OSS não existir.
- Se o caminho for uma sintaxe restrita:
  - Retorna um número inteiro, o tamanho da memória e o JSON valor em bytes.
  - Retorna null se a chave Valkey ou Redis não OSS existir.

## FIELDS

- Se o caminho for uma sintaxe aprimorada:
  - Retorna uma matriz de números inteiros que representam o número de campos de JSON valor em cada caminho.
  - Retorna uma matriz vazia se a chave Valkey ou Redis OSS não existir.
- Se o caminho for uma sintaxe restrita:
  - Retorna um número inteiro, o número de campos do JSON valor.
  - Retorna null se a chave Valkey ou Redis não OSS 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 os JSON valores no caminho em uma chave do documento. Se o caminho for a raiz, é equivalente a excluir a chave do Valkey ou do Redis. OSS

### Sintaxe

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

- **chave (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. Assumirá o padrão da raiz se não for fornecido.

### Return

- Número de elementos excluídos.
- 0 se a chave Valkey ou Redis OSS não existir.
- 0 se o JSON caminho 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 serializado JSON em um ou vários caminhos.

### Sintaxe

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

- chave (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- INDENT/NEWLINE/SPACE(opcional) — Controla o formato da JSON string retornada, ou seja, “impressão bonita”. O valor padrão é uma string vazia. Ele pode ser anulado em qualquer combinação. Eles podem ser especificados em qualquer ordem.
- NOESCAPE- Opcional, pode estar presente para compatibilidade com o legado e não tem outro efeito.
- caminho (opcional) — Zero ou mais JSON caminhos, o padrão é a 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:

- Retorna um JSON objeto em sequência de caracteres, 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.*
["\t21 2nd Street\t","\tNew York\t","\tNY\t","\t10021-3100\t"]
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
["\n\t\t21 2nd Street\t","\n\t\tNew York\t","\n\t\tNY\t","\n\t\t10021-3100\t\n"]
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
{"\$.firstName\":[\tJohn\t],"\$.lastName\":[\tSmith\t],"\$.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 $.*
["{ },{\t"a\t":1},{\t"a\t":1,\t"b\t":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

É serializado JSONs no caminho a partir de várias chaves do documento. Ele retorna null para uma chave ou caminho inexistente. JSON

### Sintaxe

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

- chave (obrigatória) — Uma ou mais OSS chaves Valkey ou Redis do tipo de documento.
- caminho (obrigatório) — Um JSON caminho.

### Return

- Matriz de strings em massa. O tamanho da matriz é igual ao número de chaves no comando. Cada elemento da matriz é preenchido com (a) o serializado JSON 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 JSON chave, o comando retornará um WRONGTYPE erro.

### 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.NUMINCRBY

Incrementa os valores numéricos no caminho por um determinado número.

### Sintaxe

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

- **chave (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.



- caminho (obrigatório) — Um JSON caminho.
- 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.
- `OVERFLOW` erro se o resultado estiver fora do intervalo de 64 bits IEEE duplos.
- `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.
- `OVERFLOW` erro se o resultado estiver fora do intervalo de 64 bits IEEE duplos.
- `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\":[],\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

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

```

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

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
```

```
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```

## JSON.NUMMULTBY

Multiplica os valores numéricos no caminho por um determinado número.

### Sintaxe

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

- chave (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- caminho (obrigatório) — Um JSON caminho.
- 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.
- `OVERFLOW` erro se o resultado estiver fora do intervalo de um número de ponto flutuante de precisão IEEE dupla 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.
- `OVERFLOW` erro se o resultado estiver fora do intervalo de um IEEE duplo 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.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[2,4,6]}"

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

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

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

```

"[null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.b.* 2
"[null,2]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.c.* 2
"[null,null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.d.* 2
"[2,null,6]"

```

### 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\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .c.* 2
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":2,\"b\":4,\"c\":6}}"

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

```

## JSON.OBJLEN

Obtém o número de chaves nos valores do objeto no caminho.

### Sintaxe

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

- **chave (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. 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.
- WRONGTYPE 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 (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. 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.
- `WRONGTYPE` 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.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 JSON valor no caminho fornecido no Protocolo de OSS Serialização Valkey ou Redis ().  
RESP Se o valor for container, a resposta será uma RESP matriz ou matriz aninhada.

- JSONnull é mapeado para a string em massa RESP nula.
- JSONOs valores booleanos são mapeados para as respectivas cadeias de caracteres RESP simples.
- Os números inteiros são mapeados para RESP números inteiros.
- Números de ponto flutuante IEEE duplo de 64 bits são mapeados para sequências de caracteres RESP em massa.
- JSONas cadeias de caracteres são mapeadas para cadeias de caracteres RESP em massa.
- JSONmatrizes são representadas como RESP Matrizes, onde o primeiro elemento é a string simples [, seguida pelos elementos da matriz.
- JSONos objetos são representados como RESP matrizes, em que o primeiro elemento é a string simples {, seguida por pares de valores-chave, cada um dos quais é uma RESP string em massa.

### Sintaxe

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

- chave (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- path (opcional) — Um JSON caminho. 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 RESP forma do valor em um caminho.
- Matriz vazia se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz que representa a RESP forma do valor no 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 JSON valores no caminho.

Se o caminho exigir um membro do objeto:

- Se o elemento pai não existir, o comando retornará um NONEXISTENT erro.
- 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 um NONEXISTENT erro.
  - Se o membro existir, seu valor será substituído pelo JSON valor.

Se o caminho exigir um índice de matriz:

- Se o elemento pai não existir, o comando retornará um NONEXISTENT erro.
- 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 OUTFBOUNDARIES erro.
- Se o elemento pai existir e o índice for válido, o elemento será substituído pelo novo JSON valor.

Se o caminho exigir um objeto ou matriz, o valor (objeto ou matriz) será substituído pelo novo JSON valor.

### Sintaxe

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

[NX | XX] Onde é possível ter 0 ou 1 de identificadores [NX | XX].

- **chave** (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **caminho** (obrigatório) — Um JSON caminho. Para uma nova chave, o JSON caminho 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

Acrescenta uma string às JSON strings no caminho.

### Sintaxe

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

- **chave** (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path** (opcional) — Um JSON caminho. Assumirá o padrão da raiz se não for fornecido.
- **json\_string** (obrigatório) — A JSON representação de uma string. Observe que uma JSON string 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.
- SYNTAXERerro se o argumento json de entrada não for uma JSON string válida.
- NONEXISTENT erro 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.
- WRONGTYPEerro se o argumento json de entrada não for uma JSON string válida.
- NONEXISTENT erro 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.STRLEN

Obtém os comprimentos dos valores da JSON string no caminho.

### Sintaxe

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

- **chave** (obrigatória) — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path** (opcional) — Um JSON caminho. 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

Altera valores booleanos entre verdadeiro e falso no caminho.

### Sintaxe

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

- **chave (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. 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 (obrigatória)** — Uma chave Valkey ou Redis do OSS tipo JSON documento.
- **path (opcional)** — Um JSON caminho. 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 JSON caminho 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-los.

### 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 em uma chave e um valor opcional, ambos definidos por você. 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 da . 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. Se você excluir um recurso, todas as tags do recurso 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 o AWS Management Console AWS CLI, o e ElastiCache API o.

Se você estiver usando IAM, você pode controlar quais usuários da sua AWS 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 AWS Management Console, 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 Amazon ElastiCache API AWS CLI, a ou uma AWS SDK, você pode usar o Tags parâmetro na ElastiCache API ação relevante 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 o ElastiCache API AWS CLI, o 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

Compatível com tags	Oferece suporte à marcação na criação
Sim	Sim
Sim	Sim
Sim	Sim

 Note

Não é possível marcar Datastores globais.

Você pode aplicar permissões de nível de recurso baseadas em tags em suas IAM políticas às ElastiCache API ações 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 de cache no grupo de replicação. Se o cluster de cache 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 de cache primário serão adicionadas ao grupo de replicação e propagadas para todos os clusters de cache.

- 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 de cache.

- `CreateCacheCluster` :

- Se o `--replication-group-id` for fornecido:

Se as tags forem incluídas na solicitação, o cluster de cache será marcado somente com essas tags. Se nenhuma tag for incluída na solicitação, o cluster de cache herdará as tags de grupo de replicação em vez das tags do cluster de cache primário.

- Se o `--snapshot-name` for fornecido:

Se as tags forem incluídas na solicitação, o cluster de cache será marcado somente com essas tags. Se nenhuma tag for incluída na solicitação, as tags de snapshot serão adicionadas ao cluster de cache.

- `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 adicionadas/removidas 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.
- Tamanho máximo da chave — 128 caracteres Unicode em UTF -8.
- Tamanho 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 tags 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

- Criação de um cache 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 instantâneos 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. Permitir a `AddTagsToResource` ação em um cluster somente se o cluster tiver a tag `Project=XYZ`.

```
{
  "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)`.

```
{
  "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).

```

{
  "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`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CreateReplicationGroup",
      "Resource": [

```

```

        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Project": "Foo"
        }
    }
}
]
}

```

5. Negar a CopySnapshot ação se o snapshot de origem tiver a tag Project= XYZ e a tag de solicitação for Service=Elasticache.

```

{
  "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.

```

{
  "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 obter mais informações, 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 ElastiCache autoprojeto. 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 etiquetas de alocação de custos são aplicadas aos ElastiCache recursos especificados em CLI e API às operações como um ARN. O tipo de recurso será um "cluster".

AmostraARN: `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.

## Gerenciando suas etiquetas de alocação de custos usando o AWS CLI

Você pode usar o AWS CLI para adicionar, modificar ou remover tags de alocação de custos.

As etiquetas de alocação de custos são aplicadas aos ElastiCache clusters. O cluster a ser marcado é especificado usando um ARN (Amazon Resource Name).

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### Tópicos

- [Listando tags usando o AWS CLI](#)
- [Adicionando tags usando o AWS CLI](#)
- [Modificando tags usando o AWS CLI](#)
- [Removendo tags usando o AWS CLI](#)

## Listando tags usando o AWS CLI

Você pode usar as tags AWS CLI para listar em um ElastiCache recurso existente usando o [list-tags-for-resource](#) operação.

O código a seguir usa o 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 o AWS CLI para listar as tags no nó Valkey ou Redis `my-cluster-001` no OSS `my-cluster` 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 vazia TagList.

```
{  
  "TagList": []  
}
```

Para obter mais informações, consulte o AWS CLI para ElastiCache [list-tags-for-resource](#).

## Adicionando tags usando o AWS CLI

Você pode usar o AWS CLI para adicionar tags a um ElastiCache recurso existente usando o [add-tags-to-resource](#) CLI operação. 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 o AWS CLI para adicionar as chaves Service e Region com os valores elasticache eus-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

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 o AWS CLI para ElastiCache [add-tags-to-resource](#).

Você também pode usar o AWS CLI para adicionar tags a um cluster ao criar um novo cluster usando a operação [create-cache-cluster](#). Você não pode adicionar tags ao criar um cluster usando



o console ElastiCache de gerenciamento. Depois que o cluster é criado, você pode usar o console para adicionar tags ao cluster.

## Modificando tags usando o AWS CLI

Você pode usar o AWS CLI para modificar as tags em um ElastiCache cluster.

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.

## Removendo tags usando o AWS CLI

Você pode usar o AWS CLI para remover tags de um cluster existente ElastiCache (Memcached) usando o [remove-tags-from-resource](#) operação.

Para o Memcached, o código a seguir usa o AWS CLI para remover as tags com as chaves Service e Region do nó no cluster `my-cluster-001` `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, o código a seguir usa o AWS CLI para remover as tags 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
```

```
--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 o AWS CLI para ElastiCache [remove-tags-from-resource](#).

## Gerenciando suas etiquetas de alocação de custos usando o ElastiCache API

Você pode usar o ElastiCache API para adicionar, modificar ou remover tags de alocação de custos.

As tags de alocação de custos são aplicadas aos clusters ElastiCache do Memcached. O cluster a ser marcado é especificado usando um ARN (Amazon Resource Name).

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### Tópicos

- [Listando tags usando o ElastiCache API](#)
- [Adicionando tags usando o ElastiCache API](#)
- [Modificando tags usando o ElastiCache API](#)
- [Removendo tags usando o ElastiCache API](#)

### Listando tags usando o ElastiCache API

Você pode usar as tags ElastiCache API para listar em um recurso existente usando o [ListTagsForResource](#) operação.

Para o Memcached, o código a seguir usa o ElastiCache API 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 Redis, o código a seguir usa o ElastiCache API 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>
```

## Adicionando tags usando o ElastiCache API

Você pode usar o ElastiCache API para adicionar tags a um ElastiCache cluster existente usando o [AddTagsToResource](#) operação. 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 o ElastiCache API 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 Redis, 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

```
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 Amazon ElastiCache API Reference.

## Modificando tags usando o ElastiCache API

Você pode usar o ElastiCache API para modificar as tags em um ElastiCache cluster.

Para modificar o valor de uma tag:

- Use [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.

## Removendo tags usando o ElastiCache API

Você pode usar o ElastiCache API para remover tags de um cluster existente ElastiCache (Memcached) usando o [RemoveTagsFromResource](#) operação.

O código a seguir usa o ElastiCache API 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>
```

## Usando o Amazon ElastiCache Well-Architected Lens

Esta seção descreve o Amazon ElastiCache Well-Architected Lens, uma coleção de princípios de design e diretrizes para projetar cargas de trabalho bem arquitetadas. ElastiCache

- O ElastiCache Lens é um complemento ao [AWS Well-Architected](#) Framework.
- Cada pilar tem um conjunto de perguntas para ajudar a iniciar a discussão sobre uma ElastiCache arquitetura.
  - 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 AWS recursos que, juntos, atendem a um requisito. Os componentes interagem com outros componentes e geralmente são equivalentes a um serviço em arquiteturas de microsserviços.

- [Carga de trabalho](#): 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 ElastiCache sem servidor e o novo mecanismo Valkey.

## Tópicos

- [Pilar de ElastiCache excelência operacional da Amazon Well-Architected Lens](#)
- [Pilar de segurança do Amazon ElastiCache Well-Architected Lens](#)
- [Pilar de confiabilidade da ElastiCache lente Amazon Well-Architected](#)
- [Pilar de ElastiCache eficiência de desempenho da lente Amazon Well-Architected](#)
- [Pilar de otimização de ElastiCache custos do Amazon Well-Architected Lens](#)

## Pilar de ElastiCache excelência operacional da Amazon Well-Architected Lens

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

- [OE 1: Como você entende e responde aos alertas e eventos acionados pelo seu ElastiCache cluster?](#)
- [OE 2: Quando e como você escala seus ElastiCache clusters existentes?](#)
- [OE 3: Como você gerencia seus recursos de ElastiCache cluster e mantém seu cluster up-to-date?](#)
- [OE 4: Como você gerencia as conexões dos clientes com seus ElastiCache clusters?](#)
- [OE 5: Como você implanta ElastiCache componentes para uma carga de trabalho?](#)
- [EO 6: Como você planeja e mitiga falhas?](#)

- [OE 7: Como você soluciona problemas de eventos do motor Valkey ou RedisOSS?](#)

## OE 1: Como você entende e responde aos alertas e eventos acionados pelo seu ElastiCache cluster?

Introdução em nível de pergunta: ao operar ElastiCache clusters, você pode, opcionalmente, receber notificações e alertas quando eventos específicos ocorrerem. ElastiCache, por padrão, registra [eventos](#) relacionados aos seus recursos, como failover, substituição de nós, operação de escalabilidade, 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 ElastiCache no ElastiCache console (depois de selecionar sua região) ou usando o [comando describe-events da Amazon Command Line Interface](#) (AWS CLI) e o [ElastiCache API](#). Configure ElastiCache para enviar notificações para eventos importantes do cluster usando o Amazon Simple Notification Service (AmazonSNS). Usar a Amazon SNS com seus clusters permite que você execute ações programáticas em eventos. ElastiCache
  - Há duas grandes categorias de eventos: eventos atuais e programados. A lista de eventos atuais inclui: criação e exclusão de recursos, operações de escalabilidade, failover, reinicialização do nó, criação de instantâneo, modificação dos parâmetros do cluster, renovação do certificado CA, eventos de falha (falha no provisionamento do cluster - VPC ou ENI -, falhas de escalabilidade - e falhas no instantâneo). ENI 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 RedisOSS)
  - [Recursos]:

- [Gerenciando SNS notificações ElastiCache da Amazon](#)
  - [Notificações de eventos e Amazon SNS](#)
- [Melhor] Para automatizar as respostas a eventos, aproveite os recursos de AWS produtos e serviços, como SNS as Funções 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 CloudWatch as métricas da Amazon para monitorar seus clusters.
- [Recursos]: [Monitore ElastiCache \(RedisOSS\) \(modo de cluster desativado\) leia endpoints de réplica usando AWS Lambda, Amazon Route 53 e Amazon SNS](#) para um caso de uso que usa Lambda e. SNS

## OE 2: Quando e como você escala seus ElastiCache clusters existentes?

Introdução em nível de pergunta: dimensionar corretamente seu ElastiCache cluster é um ato de equilíbrio que precisa ser avaliado sempre que houver alterações nos tipos de carga de trabalho 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 a sobreutilização ou a subutilização de seus recursos, ElastiCache pode escalar 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] CPU e a sobreutilização da rede nos nós primários devem ser resolvidas 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 OSS cliente Valkey ou Redis conectando-se ao endpoint do ElastiCache leitor para o modo de cluster desativado ou usando o READONLY comando para o modo de cluster ativado.

[Recursos]:

- [Encontrando pontos de extremidade de conexão em ElastiCache](#)
- [Dimensionar um cluster corretamente](#)
- [READONLYComando](#)



- [Obrigatório] Monitore a utilização de recursos essenciais do cluster CPU, como 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. Para o modo de cluster ElastiCache (RedisOSS) desativado, os nós primários e de réplica podem ser escalados 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]:

- [Monitorando as melhores práticas com ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#)
  - [Clusters de escalabilidade ElastiCache \(RedisOSS\)](#)
  - [Dimensionamento ElastiCache para clusters 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 CloudWatch métricas para verificar intervalos de tempo maiores. O aprendizado da observação de longos períodos de CloudWatch métricas deve informar sua previsão sobre a utilização dos recursos do cluster. CloudWatch pontos de dados e métricas estão disponíveis por até 455 dias.

[Recursos]:

- [Monitoramento ElastiCache \(RedisOSS\) com métricas CloudWatch](#)
  - [Monitorando o Memcached com métricas CloudWatch](#)
  - [Monitorando as melhores práticas com ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#)
- [Melhor] Se seus ElastiCache recursos forem criados com, CloudFormation é uma prática recomendada realizar alterações usando CloudFormation modelos para preservar a consistência operacional e evitar alterações de configuração não gerenciadas e desvios de pilha.

[Recursos]:

- [ElastiCache referência de tipo de recurso para CloudFormation](#)
- [Melhor] Automatize suas operações de escalabilidade usando dados operacionais de cluster e defina limites CloudWatch para configurar alarmes. Use CloudWatch Events e Simple Notification Service (SNS) para acionar funções Lambda e executar e ElastiCache API escalar seus 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]:**

- [Usando os CloudWatch alarmes da Amazon](#)
- [O que são CloudWatch eventos da Amazon?](#)
- [Usando AWS Lambda com o Amazon Simple Notification Service](#)
- [ElastiCacheAPIReferência](#)

### OE 3: Como você gerencia seus recursos de ElastiCache cluster e mantém seu cluster up-to-date?

Introdução em nível de pergunta: Ao operar em grande escala, é essencial que você seja capaz de identificar e identificar todos os seus recursos. ElastiCache Ao implementar novos recursos de aplicativos, você precisa criar simetria de versão de cluster em todos os tipos de ElastiCache ambiente: 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 ElastiCache recursos nativos permite que sua equipe de engenharia se concentre em atingir os objetivos de negócios e não na ElastiCache manutenção.

- [Melhor] Execute a versão mais recente do mecanismo disponível e aplique as atualizações de autoatendimento assim que elas estiverem disponíveis. 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 na Amazon ElastiCache](#)
- [Página de ajuda da Amazon ElastiCache Managed Maintenance and Service Updates](#)
- [Melhor] Organize seus ElastiCache recursos usando tags. 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](#)
- [ElastiCache referência de tipo de recurso para CloudFormation](#)
- [Grupos de parâmetros](#)

## OE 4: Como você gerencia as conexões dos clientes com seus ElastiCache clusters?

Introdução em nível de pergunta: Ao operar em grande escala, você precisa entender como seus clientes se conectam ao ElastiCache cluster para gerenciar os aspectos operacionais do seu aplicativo (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, lembre-se de que, ao separar as gravações das leituras, você perderá a capacidade de ler uma chave imediatamente após gravá-la devido à natureza assíncrona da replicação do Valkey e do Redis. O `WAIT` comando pode ser usado para melhorar 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 desempenho. O uso de nós de réplica para operações de leitura pode ser configurado em sua biblioteca cliente ElastiCache (RedisOSS) usando o endpoint do ElastiCache leitor para o modo de cluster desativado. Para ativar o modo de cluster, use o `READONLY` comando ElastiCache (RedisOSS). Para muitas das bibliotecas de cliente ElastiCache (RedisOSS), o ElastiCache (RedisOSS) `READONLY` é implementado por padrão ou por meio de uma definição de configuração.

[Recursos]:

- [Encontrando pontos de extremidade de conexão em ElastiCache](#)
- [READONLY](#)
- [Obrigatório] Use o agrupamento de conexões. Estabelecer uma TCP conexão tem um custo em CPU tempo, tanto do lado do cliente quanto 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 pool de conexões por meio de sua biblioteca cliente ElastiCache (RedisOSS) (se suportada), com uma estrutura disponível para seu ambiente de aplicativos, ou criá-la 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 CPU utilização necessárias para abrir e fechar conexões e realizar um TLS handshake se TLS estiver habilitado no cluster.

[Recursos]: [Configure ElastiCache \(RedisOSS\) para maior disponibilidade](#)

- [Bom] Usar pipelines (quando seus casos de uso permitirem) pode aumentar significativamente a performance.
  - Com o pipeline, você reduz o tempo de ida e volta (RTT) entre os clientes do aplicativo 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](#)

## OE 5: Como você implanta ElastiCache componentes para uma carga de trabalho?

Introdução no nível da pergunta: ElastiCache os ambientes podem ser implantados manualmente por meio do AWS console ou programaticamente por meio de kits de ferramentas APIs etcCLI. As práticas recomendadas de Excelência operacional sugerem que as implantações sejam automatizadas por meio de código sempre que possível. Além disso, ElastiCache os clusters podem ser isolados por carga de trabalho ou combinados para fins de otimização de custos.

Benefício em nível de pergunta: escolher o mecanismo de implantação mais adequado para seus ElastiCache ambientes 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 carga de trabalho, você pode optar por ter ElastiCache ambientes dedicados por carga de trabalho ou combinar várias cargas de trabalho em clusters únicos ou combinações dos mesmos. Compreender as vantagens e desvantagens pode ajudar a encontrar um equilíbrio entre Excelência operacional e Otimização de custos.

- [Obrigatório] Compreenda as opções de implantação disponíveis e automatize esses procedimentos sempre que possível. ElastiCache As possíveis vias de automação incluem CloudFormationSDK, AWS CLI/e. APIs

[Recursos]:

- [Referência de tipo ElastiCache de recurso da Amazon](#)
- [elasticache](#)
- [ElastiCache APIReferência da Amazon](#)
- [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 controle mais refinado sobre acesso, dimensionamento, escalabilidade e gerenciamento de ElastiCache recursos por carga de trabalho.
  - [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 em nível de pergunta: A excelência operacional inclui a antecipação de falhas por meio da realização de exercícios regulares “pre-mortem” para identificar possíveis fontes de falha para que possam ser removidas ou mitigadas. ElastiCache oferece um failover API que permite simular eventos de falha de nó, 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] Execute regularmente testes de failover em contas de desenvolvimento/teste.

[TestFailover](#)

## OE 7: Como você soluciona problemas de eventos do motor Valkey ou RedisOSS?

**Introdução em nível de pergunta:** a excelência operacional exige a capacidade de investigar informações de nível de serviço e de motor para analisar a integridade e o status de seus clusters. ElastiCache pode emitir registros do OSS mecanismo Valkey ou Redis para o Amazon CloudWatch e o Amazon Kinesis Data Firehose.

**Benefício em nível de pergunta:** habilitar os registros OSS do mecanismo Valkey ou Redis em ElastiCache clusters fornece uma visão dos eventos que afetam a integridade e o desempenho dos clusters. Os registros OSS do mecanismo Valkey ou Redis fornecem dados diretamente do mecanismo que não estão disponíveis por meio do mecanismo de ElastiCache eventos. Por meio da observação cuidadosa dos ElastiCache eventos (consulte o OE-1 anterior) e dos registros do motor, é possível determinar a ordem dos eventos durante a solução de problemas, tanto do ponto de vista do ElastiCache serviço quanto do ponto de vista do motor.

- [Obrigatório] Certifique-se de que a funcionalidade de registro OSS do mecanismo Redis esteja ativada, disponível a partir do ElastiCache (RedisOSS) 6.2 e versões mais recentes. 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 OSS para os registros do mecanismo Redis.
  - Selecione um registro de destino apropriado em um CloudWatch ou no Kinesis Data Firehose para manter os registros. 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: [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?](#)

- [Melhor] Se estiver usando o Amazon CloudWatch Logs, considere usar o Amazon CloudWatch Logs Insights para consultar o log do OSS mecanismo Valkey ou Redis para obter informações importantes.

Como exemplo, crie uma consulta no grupo CloudWatch Log que contém os registros do OSS mecanismo Valkey ou Redis que retornarão eventos com um LogLevel 'WARNING', como:

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Recursos]: [análise de dados de registro com o CloudWatch Logs Insights](#)

## Pilar de segurança do Amazon ElastiCache Well-Architected Lens

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

- [SEC1: Quais etapas você está tomando para controlar o acesso autorizado aos ElastiCache dados?](#)
- [SEC2: Seus aplicativos exigem autorização adicional para além ElastiCache dos controles baseados em rede?](#)
- [SEC3: Existe o risco de que os comandos possam ser executados inadvertidamente, causando perda ou falha de dados?](#)
- [SEC4: Como você garante a criptografia de dados em repouso com ElastiCache](#)
- [SEC5: Como você criptografa dados em trânsito? ElastiCache](#)
- [SEC6: Como você restringe o acesso aos recursos do avião de controle?](#)
- [SEC7: Como você detecta e responde aos eventos de segurança?](#)

## SEC1: Quais etapas você está tomando para controlar o acesso autorizado aos ElastiCache dados?

Introdução em nível de pergunta: Todos os ElastiCache clusters são projetados para serem acessados a partir de instâncias do Amazon Elastic Compute Cloud em uma VPC, funções sem servidor (AWS Lambda) ou contêineres (Amazon Elastic Container Service). O cenário mais encontrado é acessar um ElastiCache cluster de uma instância do Amazon Elastic Compute Cloud dentro da mesma Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Antes de se conectar a um cluster a partir de uma EC2 instância da Amazon, você deve autorizar a EC2 instância da Amazon a acessar o cluster. Para acessar um ElastiCache cluster em execução em um VPC, é necessário conceder a entrada de rede ao cluster.

Benefício em nível de pergunta: a entrada da rede no cluster é controlada por meio VPC de grupos de segurança. Um grupo de segurança atua como um firewall virtual para suas EC2 instâncias da Amazon para 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 de 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, ElastiCache está configurado para ser implantado exclusivamente em sub-redes privadas, de forma que elas só possam ser acessadas por meio da rede privada VPC do.

- [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 ElastiCache. Para habilitar isso, configure uma regra de entrada no grupo de segurança especificando o endereço/intervalo IP de origem, o TCP tipo de tráfego e a porta do seu ElastiCache cluster (porta padrão 6379 para (Redis), por exemplo ElastiCache ). Embora seja possível permitir um conjunto muito amplo de fontes de entrada, como todos os recursos em um VPC (0.0.0.0/0), é recomendável ser o mais granular possível na definição das regras de entrada, como autorizar somente o acesso de entrada a clientes Valkey ou Redis OSS executados em instâncias da Amazon associadas a um grupo de segurança específico. EC2

[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](#)



- AWS Identity and Access Management As políticas [obrigatórias] podem ser atribuídas a AWS Lambda funções que lhes permitem acessar ElastiCache dados. Para ativar esse recurso, crie uma função de IAM execução com a AWSLambdaVPCAccessExecutionRole permissão e, em seguida, atribua a função à AWS Lambda função.

[Recursos]: Configurando uma função Lambda para acessar a Amazon em uma ElastiCache VPC Amazon: [Tutorial: Configurando uma função Lambda para ElastiCache](#) acessar a Amazon em uma Amazon VPC

## SEC2: Seus aplicativos exigem autorização adicional para além ElastiCache dos controles baseados em rede?

Introdução em nível de pergunta: em cenários em que é necessário restringir ou controlar o acesso aos clusters ElastiCache (RedisOSS) em um nível de cliente individual, é recomendável autenticar por meio do comando ElastiCache (Redis). OSS AUTH ElastiCache Os tokens de autenticação (RedisOSS), com gerenciamento opcional de usuários e grupos de usuários, permitem que o ElastiCache (RedisOSS) exija uma senha antes de permitir que os clientes executem comandos e acessem chaves, melhorando assim a segurança do plano de dados.

Benefício em nível de pergunta: para ajudar a manter seus dados seguros, o ElastiCache (RedisOSS) fornece mecanismos de proteção contra o acesso não autorizado aos seus dados. Isso inclui impor o controle de acesso baseado em funções (RBAC) AUTH ou o AUTH token (senha) a serem usados pelos clientes para se conectarem ElastiCache antes de executar os comandos autorizados.

- [Melhor] Para ElastiCache (RedisOSS) 6.x e superior, defina controles de autenticação e autorização definindo grupos de usuários, usuários e cadeias de acesso. Atribua usuários a grupos de usuários, depois atribua grupos de usuários a clusters. Para ser utilizado RBAC, ele deve ser selecionado na criação do cluster e a criptografia em trânsito deve estar ativada. Certifique-se de usar um OSS cliente Valkey ou Redis que ofereça suporte TLS para poder aproveitar. RBAC

[Recursos]:

- [RBAC Inscrevendo-se em um grupo de replicação para ElastiCache \(RedisOSS\)](#)
- [Especificação de permissões usando uma string de acesso](#)
- [ACL](#)
- [Versões suportadas ElastiCache \(RedisOSS\)](#)

- [Melhor] Para versões ElastiCache (RedisOSS) anteriores à 6.x, além de definir um token/senha forte e manter uma política de senha rígida para ElastiCache (RedisOSS)AUTH, é uma prática recomendada alternar a senha/token. ElastiCache pode gerenciar até dois (2) tokens de autenticação a qualquer momento. Você também pode modificar o cluster para exigir explicitamente o uso de tokens de autenticação.

[Recursos]: [modificando o AUTH token em um cluster existente ElastiCache \(Redis OSS\)](#)

**SEC3: Existe o risco de que os comandos possam ser executados inadvertidamente, causando perda ou falha de dados?**

Introdução em nível de pergunta: há vários OSS comandos do Valkey ou do Redis 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 FLUSHALL comando 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 em nível de pergunta: a partir do ElastiCache (RedisOSS) 5.0.3, você pode renomear determinados comandos que podem prejudicar sua carga de trabalho. Renomear os comandos pode ajudar a evitar que sejam executados acidentalmente no cluster.

- [Obrigatório]

[Recursos]:

- [ElastiCache \(RedisOSS\) versão 5.0.3 \(obsoleta, use a versão 5.0.6\)](#)
- [Alterações nos parâmetros do Redis OSS 5.0.3](#)
- [Segurança do Redis OSS](#)

**SEC4: Como você garante a criptografia de dados em repouso com ElastiCache**

Introdução em nível de pergunta: Embora o ElastiCache (RedisOSS) seja um armazenamento de dados na 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 em nível de pergunta: ElastiCache (RedisOSS) fornece criptografia opcional em repouso para aumentar a segurança dos dados.

- [Obrigatório] A criptografia em repouso só pode ser ativada em um ElastiCache cluster (grupo de replicação) quando é criada. Um cluster existente não pode ser modificado para começar a criptografar dados em repouso. Por padrão, ElastiCache fornecerá e gerenciará as chaves usadas na criptografia em repouso.

[Recursos]:

- [Restrições de criptografia em repouso](#)
- [Ativar criptografia em repouso](#)
- [Melhor] Aproveite os tipos de EC2 instância da Amazon 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, AWS Key Management Service (KMS) pode ser usado para autogerenciar chaves mestras de cliente (). CMK Por meio da ElastiCache integração com AWS Key Management Service, você pode criar, possuir e gerenciar as chaves usadas para criptografia de dados em repouso para seu cluster ElastiCache (RedisOSS).

[Recursos]:

- [Usando chaves gerenciadas pelo cliente de AWS Key Management Service](#)
- [AWS Serviço de gerenciamento de chaves](#)
- [AWS KMSconceitos](#)

## SEC5: Como você criptografa dados em trânsito? 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 aplicativos e nós de cluster. ElastiCache (RedisOSS) suporta 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 em nível de pergunta: a criptografia ElastiCache em trânsito da Amazon é um recurso opcional que permite aumentar a segurança de seus dados nos pontos mais vulneráveis, quando eles estão em trânsito de um local para outro.

- [Obrigatório] A criptografia em trânsito só pode ser habilitada em um cluster ElastiCache (RedisOSS) (grupo de replicação) após a 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 sua carga de trabalho antes e depois da ativação. encryption-in-transit

[Recursos]:

- [Visão geral da criptografia em trânsito](#)

## SEC6: Como você restringe o acesso aos recursos do avião de controle?

Introdução em nível de pergunta: IAM políticas e ARN habilite controles de acesso refinados para ElastiCache (RedisOSS), permitindo um controle mais rígido para gerenciar a criação, modificação e exclusão de clusters (Redis). ElastiCache OSS

Benefício em nível de pergunta: o gerenciamento de ElastiCache recursos da Amazon, como grupos de replicação, nós etc. pode ser restrito a AWS contas que tenham permissões específicas com base em IAM políticas, melhorando a segurança e a confiabilidade dos recursos.

- [Obrigatório] Gerencie o acesso aos ElastiCache recursos da Amazon atribuindo AWS Identity and Access Management políticas específicas aos AWS usuários, permitindo um controle mais preciso sobre quais contas podem realizar quais ações nos clusters.

[Recursos]:

- [Visão geral do gerenciamento de permissões de acesso aos seus ElastiCache recursos](#)
- [Usando políticas baseadas em identidade \(IAMpolíticas\) para a Amazon ElastiCache](#)

## SEC7: Como você detecta e responde aos eventos de segurança?

Introdução em nível de pergunta: ElastiCache, quando implantado com RBAC ativado, exporta CloudWatch métricas para notificar os usuários sobre eventos de segurança. Essas métricas ajudam a identificar tentativas malsucedidas de autenticação, acesso a chaves ou execução de comandos para os quais RBAC os usuários de conexão não estão autorizados.

Além disso, AWS os recursos de produtos e serviços ajudam a proteger sua carga de trabalho geral automatizando implantações e registrando todas as ações e modificações para posterior revisão/ auditoria.

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 CloudWatch métricas publicadas relacionadas a falhas de RBAC autenticação e autorização.
  - AuthenticationFailures = Tentativas falhadas de autenticação no Valkey ou no Redis OSS
  - KeyAuthorizationFailures = Tentativas fracassadas dos usuários de acessar as chaves sem permissão
  - CommandAuthorizationFailures = Tentativas falhadas dos usuários de executar comandos sem permissão

[Recursos]:

- [Métricas para Valkey ou Redis OSS](#)
- [Ideal] É recomendável configurar alertas e notificações sobre essas métricas e responder conforme necessário.

[Recursos]:

- [Usando CloudWatch alarmes da Amazon](#)
- [Melhor] Use o OSS ACL LOG comando Valkey ou Redis para obter mais detalhes

[Recursos]:

- [ACL LOG](#)
- [Melhor] Familiarize-se com os recursos de AWS produtos e serviços relacionados ao monitoramento, registro e análise de ElastiCache implantações e eventos

[Recursos]:

- [Registrando ElastiCache API chamadas da Amazon com AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-verificação de backup](#)
- [Monitorando o uso com CloudWatch métricas](#)

## Pilar de confiabilidade da ElastiCache lente Amazon Well-Architected

O pilar de confiabilidade se concentra nas cargas de trabalho que executam as funções pretendidas e em como se recuperar rapidamente de falhas no atendimento às demandas. Os principais tópicos

incluem projeto de sistema distribuído, planejamento de recuperação e adaptação às mudanças nos requisitos.

## Tópicos

- [REL1: Como você está dando suporte às implantações de arquitetura de alta disponibilidade \(HA\)?](#)
- [REL2: Como você está cumprindo seus objetivos de ponto de recuperação \(RPOs\)ElastiCache?](#)
- [REL3: Como você dá suporte aos requisitos de recuperação de desastres \(DR\)?](#)
- [REL4: Como você planeja eficazmente os failovers?](#)
- [REL5: Seus ElastiCache componentes foram projetados para serem escalados?](#)

## REL1: Como você está dando suporte às implantações de arquitetura de alta disponibilidade (HA)?

Introdução em nível de pergunta: Compreender a arquitetura de alta disponibilidade da Amazon ElastiCache permitirá que você opere em um estado resiliente durante eventos de disponibilidade.

Benefício em nível de pergunta: arquitetar seus ElastiCache clusters para serem resilientes a falhas garante maior disponibilidade para suas implantações. ElastiCache

- [Obrigatório] Determine o nível de confiabilidade que você precisa para seu ElastiCache cluster. 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 (Memcached) vs ElastiCache (Redis) OSS

1. ElastiCache (Memcached) não fornece nenhum mecanismo de replicação e é usado principalmente para cargas de trabalho efêmeras.
  2. ElastiCache (RedisOSS) oferece recursos de HA discutidos abaixo
- [Melhor] Para cargas de trabalho que exigem HA, use ElastiCache (RedisOSS) no modo de cluster com no mínimo duas réplicas por fragmento, mesmo para cargas de trabalho com requisitos de taxa de transferência pequenos que exigem 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 cargas de trabalho fragmentadas, um mínimo de três fragmentos fornece uma recuperação mais rápida durante eventos de failover, pois o Valkey ou o Redis OSS Cluster Protocol exige que a maioria dos nós primários esteja disponível para atingir o quorum.
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).

ElastiCache (RedisOSS) adicionou desempenho otimizado nesses nós. Como resultado, você obtém melhor performance de replicação e sincronização, resultando em maior disponibilidade geral.

5. Monitore e dimensione corretamente para lidar com os picos de tráfego previstos: sob carga pesada, o mecanismo ElastiCache (RedisOSS) pode deixar de responder, o que afeta a disponibilidade. BytesUsedForCache e DatabaseMemoryUsagePercentage são bons indicadores do uso da memória, enquanto ReplicationLag são um indicador da integridade da 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 o [Failover API antes de um evento de failover de produção](#).

[Recursos]:

- [Configure ElastiCache \(RedisOSS\) para maior disponibilidade](#)
- [Alta disponibilidade com o uso de grupos de replicação](#)

## REL2: Como você está cumprindo seus objetivos de ponto de recuperação (RPOs)ElastiCache?

Introdução em nível de pergunta: entenda a carga de trabalho para RPO embasar as decisões sobre estratégias de ElastiCache backup e recuperação.

Benefício em nível de pergunta: ter uma RPO estratégia implementada pode melhorar a continuidade dos negócios no caso de cenários de recuperação de desastres. Projetar suas políticas de backup e restauração pode ajudá-lo a atingir seus objetivos de ponto de recuperação (RPO) para seus ElastiCache dados. ElastiCache (RedisOSS) oferece recursos de snapshot que são armazenados no Amazon S3, junto 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] Compreenda e RPO documente suas ElastiCache implantações.
  - Lembre-se de que o Memcached não oferece nenhum processo de backup.
  - Analise os recursos dos recursos de ElastiCache Backup e Restauração.
- [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 recurso de backup agendado 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 para ElastiCache \(RedisOSS\)](#)
  - [Criação de backups manuais](#)
  - [Programação de backups automáticos](#)
  - [Clusters de backup e restauração ElastiCache \(RedisOSS\)](#)

### REL3: Como você dá suporte aos requisitos de recuperação de desastres (DR)?

Introdução em nível de pergunta: a recuperação de desastres é um aspecto importante de qualquer planejamento de carga de trabalho. ElastiCache (RedisOSS) oferece várias opções para implementar a recuperação de desastres com base nos requisitos de resiliência da carga de trabalho. Com o Amazon ElastiCache Global Datastore, você pode gravar em seu cluster ElastiCache (RedisOSS) em uma região e ter os dados disponíveis para serem lidos em outros dois clusters de réplicas entre regiões, permitindo leituras de baixa latência e recuperação de desastres em todas as 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 ElastiCache componentes com base nos requisitos da carga de trabalho. 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. Certifique-se de implantar com o modo de cluster ativado em arquiteturas Multi-AZ, com um mínimo de 3 disponíveis. AZs
- 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 AWS regiões usando datastores globais](#)
  - [Restauração de um backup com redimensionamento opcional do cluster](#)
  - [Minimizando o tempo de inatividade no ElastiCache \(RedisOSS\) com o Multi-AZ](#)

## REL4: Como você planeja eficazmente os failovers?

Introdução em nível de pergunta: habilitar o Multi-AZ com failovers automáticos é uma prática recomendada. ElastiCache Em certos casos, o ElastiCache (RedisOSS) 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 tanto da configuração da biblioteca cliente ElastiCache quanto da sua biblioteca cliente.

Benefício em nível de pergunta: seguir as melhores práticas para ElastiCache failovers em conjunto com sua biblioteca cliente específica ElastiCache (RedisOSS) ajuda a minimizar o possível tempo de inatividade 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 cliente ElastiCache (RedisOSS), que também permitem que você defina a frequência e o método de atualização. Cada biblioteca de cliente oferece configurações próprias e mais detalhes estão disponíveis na documentação correspondente.

[Recursos]:

- [Minimizando o tempo de inatividade no ElastiCache \(RedisOSS\) com o Multi-AZ](#)
- Analise as melhores práticas da sua biblioteca cliente ElastiCache (RedisOSS).
- [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 do Redis, bem como as CloudWatch métricas disponíveis para relatar o atraso de OSS replicação entre os nós primário e de réplica. Para casos de uso que exigem maior segurança de dados, use o WAIT comando para forçar as réplicas a reconhecer as gravações antes de responder aos clientes conectados.

[Recursos]:

- [Métricas para Valkey ou Redis OSS](#)
- [Monitorando as melhores práticas com ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#)
- [Melhor] Valide regularmente a capacidade de resposta do seu aplicativo durante o failover usando o ElastiCache Test Failover. API

[Recursos]:

- [Testando o failover automático para uma réplica de leitura na Amazon ElastiCache \(Redis\) OSS](#)
- [Teste do failover automático](#)

## REL5: Seus ElastiCache componentes foram projetados para serem escalados?

Introdução em nível de pergunta: ao compreender os recursos de escalabilidade e as topologias de implantação disponíveis, seus ElastiCache componentes podem se ajustar com o tempo para atender às mudanças nos requisitos de carga de trabalho. ElastiCache oferece escala de 4 vias: entrada/saída (horizontal) e cima/baixo (vertical).

Benefício em nível de pergunta: seguir as melhores práticas para ElastiCache implantações fornece a maior flexibilidade de escalabilidade, além de atender ao princípio da Well Architected de escalar horizontalmente 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 informações READIOPS: adicione réplicas
  - Para saber mais WRITEOPS: adicione fragmentos (escale)
  - Para mais E/S de rede: use instâncias otimizadas para rede (aumentar a escala verticalmente).
- [Melhor] Implante seus ElastiCache componentes com o modo de cluster ativado, 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] Para o modo de cluster desativado, utilize as réplicas de leitura para aumentar a capacidade geral de leitura. ElastiCache tem suporte para até 5 réplicas de leitura no modo de cluster desativado, bem como escalabilidade vertical.
- [Recursos]:
  - [Clusters de escalabilidade ElastiCache \(RedisOSS\)](#)
  - [Aumento de escala vertical on-line](#)
  - [Dimensionamento ElastiCache para clusters Memcached](#)

## 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?](#)
- [EP 5: Como você gerencia a exclusão e/ou remoção 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] Para cargas de trabalho ElastiCache (RedisOSS), renomeie comandos computacionalmente caros, como, por exemplo KEYS, para limitar a capacidade dos usuários de executar comandos de bloqueio em clusters de produção.
- ElastiCache As cargas de trabalho (RedisOSS) que executam o mecanismo 6.x 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 CLI associando os grupos de usuários a um cluster ElastiCache (RedisOSS). No Redis OSS 6, quando RBAC ativado, podemos usar "- @dangerous" e ele não permitirá comandos caros como KEYS,, MONITORSORT, etc. para esse usuário.
- Para a versão 5.x do mecanismo, renomeie os comandos usando o `rename-commands` parâmetro no grupo de parâmetros do cluster ElastiCache (RedisOSS).
- [Melhor] Analise consultas lentas e procure técnicas de otimização.
  - Para cargas de trabalho ElastiCache (RedisOSS), 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 segundos por padrão).
  - Certas consultas podem ser realizadas com mais eficiência usando estruturas de dados complexas ElastiCache (RedisOSS). 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.
  - Para cargas de trabalho ElastiCache (RedisOSS), `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](#)
  - [referência](#)

## 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 (RedisOSS) 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 as vantagens do modo de cluster ElastiCache (RedisOSS) ativado para melhorar a escalabilidade.
  - ElastiCache Os clusters (RedisOSS) (modo de cluster ativado) oferecem suporte a [operações de escalabilidade on-line](#) (entrada/saída e cima/redução) 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 no seu cluster ElastiCache (RedisOSS) (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 OSS dados multidimensionais do Valkey ou do Redis, 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. Para cargas de trabalho ElastiCache (RedisOSS), você pode detectar teclas de atalho usando `valkey-cli --hotkeys` se uma política de LFU memória máxima 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 OSS nó Valkey ou Redis em si não fornecerá essa funcionalidade) e mantenha uma lista de nomes de chave para leitura, além do nome da chave original.
- ElastiCache [com Valkey 7.2 e superior e Redis OSS versão 6 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 com Valkey e Redis OSS para maior disponibilidade](#)
  - [Encontrando pontos de extremidade de conexão em ElastiCache](#)
  - [Práticas recomendadas para balanceamento de carga](#)
  - [Refragmentação online para Valkey ou Redis OSS \(modo de cluster ativado\)](#)
  - [Cache do lado do cliente no Valkey e no 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 Valkey e Redis OSS](#)
- [Obrigatório] Meça e colete o desempenho do cache do aplicativo em conjunto com os valores de latência e CPU utilização 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 `INFO` comando ElastiCache (RedisOSS) e não incluem a rede e o tempo de E/S. Esse é apenas o tempo consumido pelo ElastiCache (RedisOSS) para processar as operações.

[Recursos]:

- [Métricas para Valkey e Redis OSS](#)
- [Monitorando as melhores práticas com ElastiCache \(RedisOSS\) usando a 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 (RedisOSS) e ElastiCache (Memcached) são compatíveis com 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 \(RedisOSS\) - Armazenamento de dados global](#)
  - [Parâmetros específicos do Memcached](#)
  - [ElastiCache \(RedisOSS\) 5.0.3 aprimora o manuseio de E/S para aumentar o desempenho](#)
  - [Métricas para Valkey e Redis OSS](#)
  - [Configure ElastiCache \(RedisOSS\) para maior disponibilidade](#)

## EP 5: Como você gerencia a exclusão e/ou remoção 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 (RedisOSS) 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 o grupo de parâmetros do cluster `maxmemory-policy` no ElastiCache (RedisOSS).
  - A política padrão `volatile-lru` libera memória ao despejar chaves com um tempo de expiração (`TTLvalor`) definido. As políticas usadas com menos frequência (LFU) e usadas menos recentemente (LRU) removem as chaves com base no uso.
  - Para cargas de trabalho do Memcached, existe uma LRU política padrão que controla os despejos 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.
  - Para cargas de trabalho ElastiCache (RedisOSS), 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 cargas de trabalho ElastiCache (RedisOSS) 6.x, o comportamento do `DEL` comando pode ser modificado no grupo de parâmetros usando o parâmetro. `lazyfree-lazy-user-del`
- [Recursos]:
  - [Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros](#)
  - [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 ElastiCache (RedisOSS) 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 dados e comando ElastiCache (RedisOSS) tem suas próprias assinaturas de desempenho 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](#)
- Os comandos [Best] ElastiCache (RedisOSS) têm uma complexidade de tempo definida pela notação Big O. Essa complexidade temporal de um comando é uma representação algorítmica/matemática 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 ElastiCache (RedisOSS), 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 GUI visibilidade do modelo de dados em seu cluster.

[Recursos]:

- [Comandante do Redis OSS](#)
- [Navegador 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 os comandos serem executados, você pode determinar quais comandos resultam em baixo desempenho, bem como os comandos que impedem que o mecanismo tenha um desempenho ideal. ElastiCache (RedisOSS) 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] Amazon ElastiCache (RedisOSS) executando a versão 6.0 do mecanismo ou mais recente, grupo de parâmetros configurado corretamente e SLOWLOG registro habilitado no cluster.
  - Os parâmetros necessários só estão disponíveis quando a compatibilidade da versão do mecanismo está definida como Valkey 7.2 e superior ou Redis OSS versão 6.0 ou superior.
  - SLOWLOGo registro ocorre quando o tempo de execução de um comando no servidor demora mais do 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 SLOWLOG para arquivar registros em armazenamento permanente ou para acionar o ajuste automático dos parâmetros do cluster.
  - Determine JSON se o TEXT formato simples atende melhor às suas necessidades.
  - Forneça IAM permissões 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 OSS mecanismo Valkey ou Redis antes de ser registrado como um comando de execução lenta. 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 de execução lenta são capturados na OSS memória Valkey ou Redis a qualquer momento. 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 do `valkey-cli` por meio do `SLOWLOG` comando, considere aumentar esse valor. Isso permite que mais comandos permaneçam na memória Valkey ou RedisOSS.

Se você estiver emitindo `SLOWLOG` os dados 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 faço para ativar o log lento em um cluster de cache ElastiCache \(RedisOSS\)?](#)
  - [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 OSS automático Valkey ou Redis, seus ElastiCache 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 (RedisOSS) O Auto Scaling monitora continuamente CPU sua utilização de /Memória para garantir que seu cluster esteja operando nos níveis de desempenho desejados.

- [Obrigatório] Ao iniciar um cluster para ElastiCache (RedisOSS):
  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]:

- [Dimensionamento de clusters no Valkey e no Redis OSS \(modo de cluster ativado\)](#)
- [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 sua aplicação com as workloads mínimas e máximas esperadas, para identificar o número exato de fragmentos/réplicas necessários para que o cluster desenvolva políticas de ajuste de escala e reduza os problemas de disponibilidade.

[Recursos]:

- [Registrar uma meta estável](#)
- [Registrando um alvo escalável usando o AWS CLI](#)

## Pilar de otimização de ElastiCache custos do Amazon Well-Architected Lens

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

- [COST1: Como você identifica e acompanha os custos associados aos seus ElastiCache recursos? Como você desenvolve mecanismos para permitir que os usuários criem, gerenciem e descartem os recursos criados?](#)
- [COST2: Como você usa ferramentas de monitoramento contínuo para ajudá-lo a otimizar os custos associados aos seus ElastiCache recursos?](#)
- [COST3: 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?](#)

**COST1: Como você identifica e acompanha os custos associados aos seus ElastiCache recursos? 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 implementados para rastrear os recursos criados desde o desenvolvimento até a produção e a aposentadoria ajuda a gerenciar os custos associados ElastiCache a.

Benefício em nível de pergunta: o rastreamento contínuo de todos os custos associados à sua carga de trabalho exige uma compreensão profunda da arquitetura que inclui 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 em Nuvem (CCoE) com um de seus estatutos fundadores para definir, monitorar e agir sobre métricas relacionadas ao uso de suas organizações ElastiCache . Se um CCoE existir e funcionar, certifique-se de que ele saiba como ler e rastrear os custos associados ElastiCache a. Quando os recursos forem criados, use IAM funções e políticas 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. CCoE deve identificar, definir e publicar métricas de custo que são atualizadas regularmente — mensalmente — sobre o ElastiCache uso de chaves 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. CCoE consiste em uma equipe multifuncional com representação não exclusiva das equipes de engenharia de software, gerenciamento de dados, equipe de produtos, finanças e equipes de liderança em sua organização.

[Recursos]:

- [Criar um Centro de Excelência da Nuvem](#)
- [ElastiCache Preços da Amazon](#)



- [Obrigatório] Use etiquetas de alocação de custos para monitorar os custos em um baixo nível de granularidade. Use o Gerenciamento de AWS Custos para visualizar, entender e gerenciar seus AWS custos e uso ao longo do tempo.
  1. Use tags para organizar seus recursos e tags de alocação de custos para monitorar seus AWS custos em um nível detalhado. Depois de ativar as tags de alocação de custos, AWS use as tags de alocação de custos para organizar seus custos de recursos em seu relatório de alocação de custos, para facilitar a categorização e o controle de seus custos. AWS AWS fornece dois tipos de tags de alocação de custos: tags AWS geradas e tags definidas pelo usuário. AWS define, cria e aplica as tags AWS geradas para você, e você define, cria e aplica tags definidas pelo 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 etiquetas de alocação de custos para organizar sua AWS fatura de forma a refletir sua própria estrutura de custos. Ao adicionar etiquetas de alocação de custos aos seus recursos na Amazon ElastiCache, você poderá rastrear os custos agrupando as despesas em suas faturas por valores de etiquetas de recursos. Considere combinar etiquetas para monitorar os custos com um maior nível de detalhes.

[Recursos]:

- [Usando tags AWS de alocação de custos](#)
- [Monitorar os custos com etiquetas de alocação de custos](#)
- [AWS Explorador de Custos](#)
- [Melhor] Conecte o ElastiCache custo 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]:

- [Monitorando o uso com CloudWatch métricas](#)
- [Bom] Mantenha a visibilidade up-to-date arquitetônica e operacional das métricas e custos em toda a carga de trabalho que usa ElastiCache.

1. Entenda todo o seu ecossistema de soluções e ElastiCache tende a fazer parte de um ecossistema completo de AWS serviços em seu conjunto de tecnologias, desde clientes até API Gateway, Redshift e QuickSight 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 computação, rede, armazenamento, políticas de ciclo de vida, coleta de métricas, bem como os ElastiCache elementos operacionais e funcionais do seu aplicativo
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, responsabilidade, priorização e recursos é crucial para que você tenha uma estratégia eficaz de gerenciamento de custos para você. ElastiCache

## COST2: Como você usa ferramentas de monitoramento contínuo para ajudá-lo a otimizar os custos associados aos seus ElastiCache recursos?

Introdução em nível de pergunta: você precisa buscar um equilíbrio adequado entre suas métricas de ElastiCache custo e desempenho do aplicativo. CloudWatch A Amazon fornece visibilidade das principais métricas operacionais que podem ajudá-lo a avaliar se seus ElastiCache recursos estão sendo utilizados em excesso ou subutilizados, em relação às suas necessidades. Do ponto de vista da otimização de custos, você precisa entender quando está superprovisionado e ser capaz de desenvolver mecanismos adequados para redimensionar seus ElastiCache recursos e, ao mesmo tempo, manter suas necessidades operacionais, de disponibilidade, resiliência e desempenho.

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 ElastiCache recursos superdimensionados por longos períodos de tempo.

- [Obrigatório] Use CloudWatch para monitorar seus ElastiCache clusters e analisar como essas métricas se relacionam com seus painéis do AWS Cost Explorer.

1. ElastiCache fornece métricas em nível de host (por exemplo, CPU uso) 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.
2. ElastiCache métricas de desempenho (CPUUtilization,, EngineUtilization SwapUsage CurrConnections, e despejos) podem indicar que você precisa aumentar ou diminuir a escala (usar tipos de nós de cache maiores/menores) ou entrada/saída (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]:

- [Monitorando o uso com CloudWatch métricas](#)
  - [Que métricas devo monitorar?](#)
  - [ElastiCachePreços da Amazon](#)
- [Obrigatório] Entenda e documente sua estratégia de backup e as implicações de custo.
    1. Com ElastiCache isso, 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 seus ElastiCache clusters usando nós sob demanda até coletar dados suficientes para estimar os requisitos de 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.

**COST3:** 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 requer um exame periódico do tempo de ElastiCache inatividade do objeto.

**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 ao SSD armazenamento disponível por instância, as instâncias de armazenamento em camadas suportam uma capacidade total de dados por instância muito maior.

- [Obrigatório] Entenda as limitações das instâncias com hierarquização de dados.
  1. Disponível somente para clusters ElastiCache (RedisOSS).
  2. Somente alguns tipos de instância oferecem suporte à hierarquização de dados.
  3. Somente a versão 6.2 e superior ElastiCache (RedisOSS) é suportada
  4. Itens grandes não são trocados por SSD Objetos com mais de 128 MiB são mantidos na memória.

[Recursos]:

- [Hierarquização de dados](#)
- [ElastiCachePreços da Amazon](#)
- [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 desempenho para acessar objetos usados com menos frequência, pois eles são trocados por locais. SSD 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 ElastiCache clusters usando instâncias que não sejam de armazenamento de dados em camadas até entender melhor seus requisitos.
    2. Analise o padrão de uso de dados do seu ElastiCache cluster.
    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.

[Recursos]:

- [OBJECT IDLETIME](#)
- [ElastiCachePreços da Amazon](#)

# 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 conselhos sobre solução de problemas e respostas a perguntas comuns de suporte, visite o [Centro de AWS Conhecimento](#)

## Tópicos

- [Problemas de conectividade](#)
- [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 persistentes de conexão](#)
- [Related Topics](#)

## Problemas de conectividade

Se você não conseguir se conectar ao ElastiCache cache, considere uma das seguintes opções:

1. Usando TLS: Se você estiver com uma conexão interrompida ao tentar se conectar ao seu ElastiCache endpoint, talvez não esteja usando TLS em seu cliente. Se você estiver usando o ElastiCache Serverless, a criptografia em trânsito estará sempre ativada. Certifique-se de que seu cliente esteja usando TLS para se conectar ao cache. [Saiba mais sobre como se conectar a um cache TLS habilitado.](#)
2. VPC: ElastiCache os caches são acessíveis somente de dentro de a. VPC Certifique-se de que a EC2 instância a partir da qual você está acessando o ElastiCache cache e o cache sejam criados na mesma VPC. Como alternativa, você deve ativar o [VPCpeering](#) entre o VPC local em que sua EC2 instância reside e o VPC local 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 o acesso às portas do cache (6379 e 6380 para servidores sem servidor e 6379, por padrão, para projetos próprios). ElastiCache usa essas portas para aceitar comandos Valkey ou RedisOSS. Saiba mais sobre como configurar o acesso à porta [aqui](#).

Se a conexão continuar difícil, [Problemas persistentes de conexão](#) consulte 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 RedisOSS. Clusters autoprojetoados podem ser acessados de clientes em qualquer um dos modos, dependendo da configuração do cluster.

Se você estiver enfrentando erros em seu cliente, considere o seguinte:

1. Modo de cluster: se você estiver enfrentando CROSSLOT erros ou erros com o [SELECT](#) comando, talvez esteja tentando acessar um cache habilitado para o modo de cluster com um OSS cliente Valkey ou Redis que não suporta o protocolo Cluster. ElastiCache O Serverless oferece suporte apenas a clientes que oferecem suporte ao protocolo de cluster Valkey ou RedisOSS. Se você quiser usar Valkey ou Redis OSS em “Modo de cluster desativado” (CMD), deverá criar seu próprio cluster.
2. CROSSLOT erros: Se você estiver enfrentando o ERR CROSSLOT Keys in request don't hash to the same slot erro, talvez esteja tentando acessar chaves que não pertencem ao mesmo slot em um cache do modo 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 OSS clientes Valkey ou Redis, 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 seu 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 AWS Health Dashboard` e o `Personal Health Dashboard` para obter mais informações. Se necessário, considere abrir um caso de suporte com AWS Support.

Considere as seguintes melhores práticas e estratégias para reduzir a latência:

- Ativar leitura da réplica: se seu aplicativo permitir, recomendamos ativar o recurso “Ler da réplica” em seu OSS cliente Valkey ou Redis 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 ativar o recurso Ler da réplica em seu cliente significa que seu aplicativo aceita uma eventual consistência nos dados. Seu aplicativo pode receber 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 ElastiCache Serverless cria endpoints nessas sub-redes. VPC Certifique-se de que seu aplicativo seja implantado no mesmo AZs. Caso contrário, seu aplicativo poderá incorrer em 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 TCP conexão TLS habilitada usando o protocolo. RESP Iniciar a conexão (incluindo a autenticação da conexão, se configurada) leva tempo, então a latência da primeira solicitação é maior do que a normal. Solicitações em uma conexão já inicializada oferecem ElastiCache baixa latência consistente. Por esse motivo, você deve considerar usar o pool de conexões Valkey ou Redis existentes. OSS
- **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 RedisOSS, 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 com solicitações limitadas.
- **Distribuição uniforme de chaves e solicitações:** ElastiCache com o Valkey e o RedisOSS, uma distribuição desigual de chaves ou solicitações por slot pode resultar em um hot slot que pode resultar em latência elevada. ElastiCache O Serverless suporta até 30.000 ECPUs /segundo (90.000 ECPUs /segundo 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, limitar a taxa na qual as API chamadas são processadas por vários componentes do serviço é chamado de limitação. 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 OSS clientes Valkey ou Redis 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 limitadas, 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:** ElastiCache com o Valkey ou o RedisOSS, 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 em um único slot exceder 30.000 por ECPUs segundo, em uma carga de trabalho que executa comandos/simples. SET GET
- **Leia da réplica:** se seu aplicativo permitir, considere usar o recurso “Ler da réplica”. A maioria dos OSS clientes Valkey ou Redis 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 atingir até 90.000 ECPUs /segundo em um único slot, para cargas de trabalho com comandos/simples. SET GET

## Problemas persistentes de conexão

Os itens a seguir devem ser verificados durante a solução de problemas persistentes de conectividade com ElastiCache:

### Tópicos

- [Grupos de segurança](#)
- [Rede ACLs](#)
- [Tabelas de rotas](#)
- [DNSresolução](#)
- [Identificação de problemas com o diagnóstico do lado do servidor](#)
- [Validação da conectividade de rede](#)
- [Limites relacionados à rede](#)
- [CPUUso](#)
- [Conexões sendo encerradas do lado do servidor](#)
- [Solução de problemas do lado do cliente para instâncias da Amazon EC2](#)
- [Dissecação do tempo necessário para concluir uma única solicitação](#)

### Grupos de segurança

Grupos de segurança são firewalls virtuais que protegem seu ElastiCache cliente (EC2instância, AWS Lambda função, ECS contêiner da Amazon etc.) e ElastiCache cache. 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 no cliente ou na ElastiCache lateral.

Você pode ter um único grupo de segurança atribuído ao mesmo tempo ao cliente e ao ElastiCache cluster, ou grupos de segurança individuais para cada um.

Em ambos os casos, você precisa permitir o tráfego de TCP saída na ElastiCache porta da origem e o tráfego de entrada na mesma porta para. 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 ElastiCache cluster em uma Amazon VPC](#).

## Rede 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. ACLsAs redes são atribuídas a sub-redes, não a recursos específicos. É possível ter o mesmo recurso ACL atribuído ao cliente ElastiCache e ao mesmo, especialmente se estiverem na mesma sub-rede.

Por padrão, a rede ACLs permite todo o tráfego. No entanto, é possível personalizá-las para negar ou permitir tráfego. Além disso, a avaliação das ACL regras é sequencial, o que significa que a regra com o menor número correspondente ao tráfego a permitirá ou negará. A configuração mínima para permitir o OSS tráfego Valkey ou Redis é:

Rede do lado do clienteACL:

- Regras de entrada:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Tipo: TCP Regra personalizada;
- Protocolo: TCP
- Port Range: 1024-65535
- Fonte: 0.0.0.0/0 (ou crie regras individuais para as sub-redes do cluster) ElastiCache
- Allow/Deny: Allow
  
- Regras de saída:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Tipo: TCP Regra personalizada;
- Protocolo: TCP
- Port Range: 6379
- Fonte: 0.0.0.0/0 (ou as sub-redes do cluster). ElastiCache Lembre-se de que o uso específico IPs pode criar problemas em caso de failover (ou escalonamento do cluster).
- Allow/Deny: Allow

## ElastiCache RedeACL:

- Regras de entrada:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Tipo: TCP Regra personalizada;
- Protocolo: TCP
- Port Range: 6379
- Fonte: 0.0.0.0/0 (ou crie regras individuais para as sub-redes do cluster) ElastiCache
- Allow/Deny: Allow
  
- Regras de saída:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Tipo: TCP Regra personalizada;
- Protocolo: TCP
- Port Range: 1024-65535
- Fonte: 0.0.0.0/0 (ou as sub-redes do cluster). ElastiCache Lembre-se de que o uso específico IPs pode criar problemas em caso de failover (ou escalonamento do cluster).
- Allow/Deny: Allow

Para obter mais informações, consulte [Rede ACLs](#).

## Tabelas de rotas

Da mesma forma que a RedeACLs, cada sub-rede pode ter tabelas de rotas diferentes. Se os clientes e o ElastiCache cluster estiverem em sub-redes diferentes, certifique-se de que suas tabelas de rotas permitam que eles se conectem.

Ambientes mais complexos, envolvendo roteamento dinâmico VPCs múltiplo ou firewalls de rede, podem se tornar difíceis de solucionar. Consulte [Validação da conectividade de rede](#) para confirmar se as configurações de rede são apropriadas.

## DNSresolução

ElastiCache fornece os endpoints do serviço com base em DNS nomes. 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.

DNSAs configurações personalizadas (ou seja, não usar o VPC DNS serviço) podem não estar ElastiCache cientes dos DNS nomes fornecidos. Certifique-se de que seu sistema possa resolver com êxito os ElastiCache endpoints usando ferramentas do sistema como `dig` (conforme mostrado a seguir) `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 do nome por meio do VPC DNS serviço:

```
$ 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

CloudWatch métricas e informações de tempo de execução do ElastiCache mecanismo são fontes ou informações comuns para identificar possíveis fontes de problemas de conexão. Uma boa análise geralmente começa com os seguintes itens:

- **CPUuso:** Valkey e Redis OSS são aplicativos multiencadeados. No entanto, a execução de cada comando ocorre em um único thread (principal). Por esse motivo, ElastiCache fornece as métricas `CPUUtilization EngineCPUUtilization` e `EngineCPUUtilization` fornece a CPU utilização dedicada ao OSS processo Valkey ou Redis e `CPUUtilization` o uso em todos. vCPUs Os nós com mais de um v CPU geralmente têm valores diferentes para `CPUUtilization` e `EngineCPUUtilization`, sendo o segundo geralmente mais alto. A alta `EngineCPUUtilization` pode ser causada por um número elevado de solicitações ou operações complexas que levam muito CPU tempo para serem concluídas. 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.

- `GetTypeCmds` e `EngineCPUUtilization`: Essas métricas correlacionadas com `GetTypeCmds` podem ajudar a entender se a carga é significativamente maior para solicitações de gravação, medida por `GetTypeCmds`, 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 desativados no modo de cluster, o uso de réplicas de leitura pode ser feito criando uma configuração de conexão adicional no aplicativo usando o endpoint do leitor. 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 corretos, a biblioteca poderá descobrir automaticamente a topologia do cluster, os nós de réplica, habilitar as operações de leitura por meio do OSS comando [READONLY](#) Valkey ou Redis e enviar as solicitações de leitura às 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 gerenciar conexões implica uma CPU sobrecarga significativa. Além disso, o aperto de mão TCP tridirecional necessário para criar novas conexões afetará negativamente os tempos gerais de resposta.

Um ElastiCache nó 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. O Valkey e o Redis OSS oferecem desempenho ideal com um pequeno número de `currConnections`. Manter `currConnection` a ordem de dezenas ou centenas minimiza o uso de recursos para oferecer suporte a conexões individuais, como buffers de clientes e CPU ciclos para atender à conexão.

- Throughput da rede

- Determine a largura de banda: ElastiCache os nós têm largura de banda de rede proporcional ao tamanho do nó. Como as aplicações têm características diferentes, os resultados podem variar de acordo com a workload. Como exemplos, aplicativos com alta taxa de pequenas solicitações tendem a afetar mais o CPU uso do que a taxa de transferência da rede, enquanto chaves maiores causam 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 melhor prática para cargas de trabalho intensivas de gravação é usar ElastiCache (RedisOSS) com o modo de cluster ativado para que as gravações possam ser balanceadas em vários fragmentos ou escalar para um tipo de nó com mais recursos de rede.
- O CloudWatchmetrics NetworkBytesIn e NetworkBytesOut fornece 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 a complexidade de seus 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 sua respectiva complexidade de tempo, acesse os comandos [Valkey e Redis OSS](#).

Exemplos de possíveis problemas:

- Scripts Lua: o Valkey e o 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 motor e são atômicos por definição, o que significa que nenhum outro comando ou script poderá ser executado enquanto um script estiver em execução. Os scripts Lua oferecem 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. Ativado ElastiCache, o tempo de execução

dos scripts Lua é limitado a 5 segundos. Scripts que não forem gravados no keypace 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. [As transações](#) são a alternativa para garantir a consistência de várias modificações importantes relacionadas no RedisOSS. 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 levarão um CPU tempo significativo se a lista de parâmetros for grande ou contiver uma lista grande, conjunto, conjunto classificado 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 DEL é UNLINK, que é um comando assíncrono disponível desde o Redis 4. OSS UNLINK deve ser preferido DEL sempre que possível. A partir do ElastiCache (RedisOSS) 6x, o `lazyfree-lazy-user-del` parâmetro faz com que o DEL comando se comporte como UNLINK quando ativado. Para obter mais informações, consulte Alterações de [parâmetros 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 lista extensa de parâmetros afetará o CPU uso.

Embora CPU a utilização por si só não seja a causa dos problemas de conectividade, gastar muito tempo processando um ou alguns comandos em várias chaves pode causar falhas em outras solicitações e aumentar a CPU utilização geral.

O número de chaves e seu tamanho afetarão a complexidade do comando e, conseqüentemente, 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 que atuam em vários tipos de dados: o Redis OSS também fornece comandos que atuam em uma ou várias chaves, independentemente do tipo de dados. ElastiCache (RedisOSS) 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 de big data: além dos hashes, outras estruturas de dados podem ser intensivas. 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 DNS resolução, grupos de segurança, rede e tabelas de rotasACLs, a conectividade pode ser validada com o Reachability VPC 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 ENI ID (Elastic Network Interface Identification) de um dos ElastiCache nós disponíveis em seuVPC. É possível fazer isso da seguinte maneira:

1. Vá para <https://console.aws.amazon.com/ec2/v2/home?#NIC:>
2. Filtre a lista de interfaces pelo nome do ElastiCache cluster ou pelo endereço IP obtido das DNS validações anteriores.
3. Anote ou salve o ENI ID. Se várias interfaces forem mostradas, revise a descrição para confirmar se elas pertencem ao ElastiCache cluster correto e escolha uma delas.
4. Prossiga para a próxima etapa.
5. Criar um caminho de análise em [https://console.aws.amazon.com/vpc/casa?#](https://console.aws.amazon.com/vpc/casa?#ReachabilityAnalyzer) ReachabilityAnalyzer e escolha as seguintes opções:
  - Tipo de fonte: escolha a instância se o seu ElastiCache cliente é executado em uma EC2 instância da Amazon ou uma interface de rede (se ele usa outro serviço, como AWS

Fargate Amazon ECS com rede awsvpc AWS Lambda, etc.), e o respectivo ID do recurso (EC2instância ou ENI ID);

- Tipo de destino: Escolha a interface de rede e selecione o Elasticache na ENI lista.
- Porta de destino: especifique 6379 para ElastiCache (RedisOSS) ou 11211 para (Memcached). ElastiCache 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 TCP conectividade na porta de ElastiCache serviço: No Amazon Linux, Nping está disponível no pacote nmap e pode testar a TCP conectividade na ElastiCache porta, 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 do ElastiCache cluster, 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 nping falharem e os testes do VPCReachability Analyzer forem aprovados, peça ao administrador do sistema que analise possíveis regras de firewall baseadas em host, regras de roteamento assimétrico ou qualquer outra restrição possível no nível do sistema operacional.

No ElastiCache console, verifique se a criptografia em trânsito está habilitada nos detalhes ElastiCache do seu cluster. Se a criptografia em trânsito estiver ativada, confirme se a TLS sessão 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 TLS negociação 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](https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS)

Se todos os testes de infraestrutura e sistema operacional tiverem sido aprovados, mas seu aplicativo ainda não conseguir se conectar ElastiCache, verifique se as configurações do aplicativo estão em conformidade com as ElastiCache configurações. Erros comuns são:

- Seu aplicativo não suporta o modo de ElastiCache cluster e ElastiCache tem o modo de cluster ativado;
- Seu aplicativo não suporta TLS/e ElastiCache tem SSL a criptografia em trânsito ativada;
- O aplicativo suporta TLS/SSL, mas não tem 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 ElastiCache nó permite até 65.000 conexões simultâneas em todos os clientes. Esse limite pode ser monitorado por meio das `CurConnections` métricas ativadas 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 (ElastiCache nó) e porta. O comando a seguir mostra quantas conexões existem para um ElastiCache nó 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 [CloudWatch métricas do Redis OSS](#) para identificar possíveis limites de rede atingidos no ElastiCache nó:
  - `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 longos períodos, provavelmente serão seguidos por atraso na replicação, aumento nos bytes usados para cache, queda na memória liberável, alta troca e uso. CPU EC2As instâncias da Amazon também têm limites de rede que podem ser rastreados por meio de [métricas de ENA drivers](#). Instâncias Linux com suporte de rede aprimorado e ENA drivers 2.2.10 ou mais recentes podem analisar os contadores de limite com o comando:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

## CPUUso

A métrica de CPU uso é o ponto de partida da investigação, e os itens a seguir podem ajudar a reduzir possíveis problemas ElastiCache paralelos:

- **Redis OSS SlowLogs:** a configuração ElastiCache padrão 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 19:26:07UTC, levou 4,8 segundos (4.823ms) para ser concluído e foi causado pelo comando solicitado ao cliente 1.2.3.4. KEYS

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 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  $O(1)$  forem frequentemente relatados, verifique os outros fatores de alto CPU uso mencionados anteriormente.

- Métricas de latência: ElastiCache (RedisOSS) fornece CloudWatch métricas 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 aos comandos Lua Script, eval, evalsha;
  - GeoSpatialBasedCmdsLatency: geodist, geohash, geopos, georadius, georadiusbymember, geoadd;



A partir do ElastiCache Redis OSS 2.8.22, 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 CPU utilização, levar a maiores tempos de resposta e, conseqüentemente, levar ao tempo limite do cliente durante sua 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 ElastiCache (RedisOSS) 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 cliente ou no ElastiCache lado. Quando uma correção imediata não é possível do lado do cliente, alguns administradores definem um valor de "tempo limite" em seu grupo de ElastiCache parâmetros. O tempo limite é o tempo em segundos permitido para que as conexões ociosas persistam. Se o cliente não enviar nenhuma solicitação no período, o OSS mecanismo do Redis 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 ElastiCache (RedisOSS) não restringem o tamanho dos buffers de saída regulares do cliente. 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 (RedisOSS) pode optar por desconectar clientes que consomem grandes buffers de saída do cliente para liberar memória e manter 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:

- `client-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 conexões que excedem o `client-output-buffer-limit-normal-soft-limit`;
- `client-output-buffer-limit-normal-hard-limit`: Uma conexão que atinja esse limite será encerrada imediatamente.

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 do lado do cliente para instâncias da Amazon EC2

A carga e a capacidade de resposta do lado do cliente também podem afetar as solicitações de ElastiCache. EC2 os limites da instância e do sistema operacional precisam ser cuidadosamente analisados ao solucionar problemas intermitentes de conectividade ou tempo limite. Alguns pontos-chave a observar:

- CPU:
  - EC2 CPU uso da instância: certifique-se de que CPU não esteja saturada ou próxima de 100 por cento. A análise histórica pode ser feita via CloudWatch, mas lembre-se de que a granularidade dos pontos de dados é de 1 minuto (com o monitoramento detalhado ativado) ou 5 minutos;
  - Se estiver usando [EC2 instâncias intermitentes](#), certifique-se de que seu saldo CPU de crédito não tenha sido esgotado. Essas informações estão disponíveis na `CPUcreditBalance` CloudWatch métrica.

- Períodos curtos de alto CPU uso podem causar tempos limite sem refletir na utilização de 100%. 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 EC2 instância da Amazon](#)
  - 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;
    - `contrack_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 da instância, via. NTP 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 PPS limite pode ser atingido quando o tráfego da rede consiste em milhares ou milhões de solicitações muito pequenas por segundo. ElastiCache o tráfego pode ser otimizado para fazer melhor uso dos pacotes de rede por meio de pipelines ou comandos que realizam várias operações ao mesmo tempo, em vez de. MGET GET

## Dissecação do tempo necessário para concluir uma única solicitação

- On the network: Tcpcdump and Wireshark (tshark na linha de comando) são ferramentas úteis para entender quanto tempo a solicitação levou para percorrer a rede, acionar o ElastiCache motor e obter uma devolução. 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, tcpcdump 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
```

Na saída acima, podemos confirmar que o handshake TCP tridirecional foi concluído em 222 microssegundos (918091 - 917869) e que 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 o mesmo OSS PING comando Redis 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\rnotls20201224\6tihew"..., 2048, 0, {sa_family=AF_INET,
sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
"\204\207\201\200\0\1\0\1\0\0\0\0\rnotls20201224\6tihew"...,
 65536, 0, {sa_family=AF_INET, sin_port=htons(53),
sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
 0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
 [128]) = -1 ENOTSOCK
(Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
 [128->0]) = 7
+PONG
```

```
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
(+ 0.003569) +++ exited with 0 +++
```

No exemplo acima, o comando levou um pouco mais de 54 milissegundos para ser concluído (752110 - 697712 = 54398 microssegundos).

Foi necessário um tempo significativo, aproximadamente 20 ms, para instanciar nc e fazer a resolução do nome (de 697712 a 717890). Depois disso, foram necessários 2 ms para criar o TCP soquete (745659 a 747858) e 0,4 ms (747858 a 748330) para enviar e receber a resposta da solicitação.

## Related Topics

- [the section called “Práticas recomendadas e estratégias de armazenamento em cache”](#)



# Segurança na Amazon ElastiCache

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar 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 à Amazon ElastiCache, consulte [AWS Services in Scope by Compliance Program](#).
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço 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.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a Amazon ElastiCache. Os tópicos a seguir mostram como configurar a Amazon ElastiCache para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam você a monitorar e proteger seus ElastiCache recursos da Amazon.

## Tópicos

- [Proteção de dados na 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 em AWS ElastiCache](#)
- [Atualizações de serviço em ElastiCache](#)
- [Vulnerabilidades e exposições comuns \(CVE\): vulnerabilidades de segurança abordadas em ElastiCache](#)

## Proteção de dados na Amazon ElastiCache

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS ElastiCache (ElastiCache). Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa toda a AWS nuvem. 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 AWS serviços que você usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#).

Para fins de proteção de dados, recomendamos que você proteja as credenciais da AWS conta e configure contas individuais com 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 a autenticação multifator (MFA) com cada conta.
- Use TLS para se comunicar com AWS os recursos.
- Configure API e registre as atividades do usuário com AWS CloudTrail.
- Use soluções AWS de criptografia, juntamente com todos os controles de segurança padrão nos AWS serviços.
- 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 ElastiCache ou outros AWS serviços usando o console, API, AWS CLI, ou AWS SDKs. Todos os dados que você inserir ElastiCache ou outros serviços podem ser coletados para inclusão nos registros de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

### Tópicos

- [Segurança de dados na Amazon ElastiCache](#)

## Segurança de dados na Amazon ElastiCache

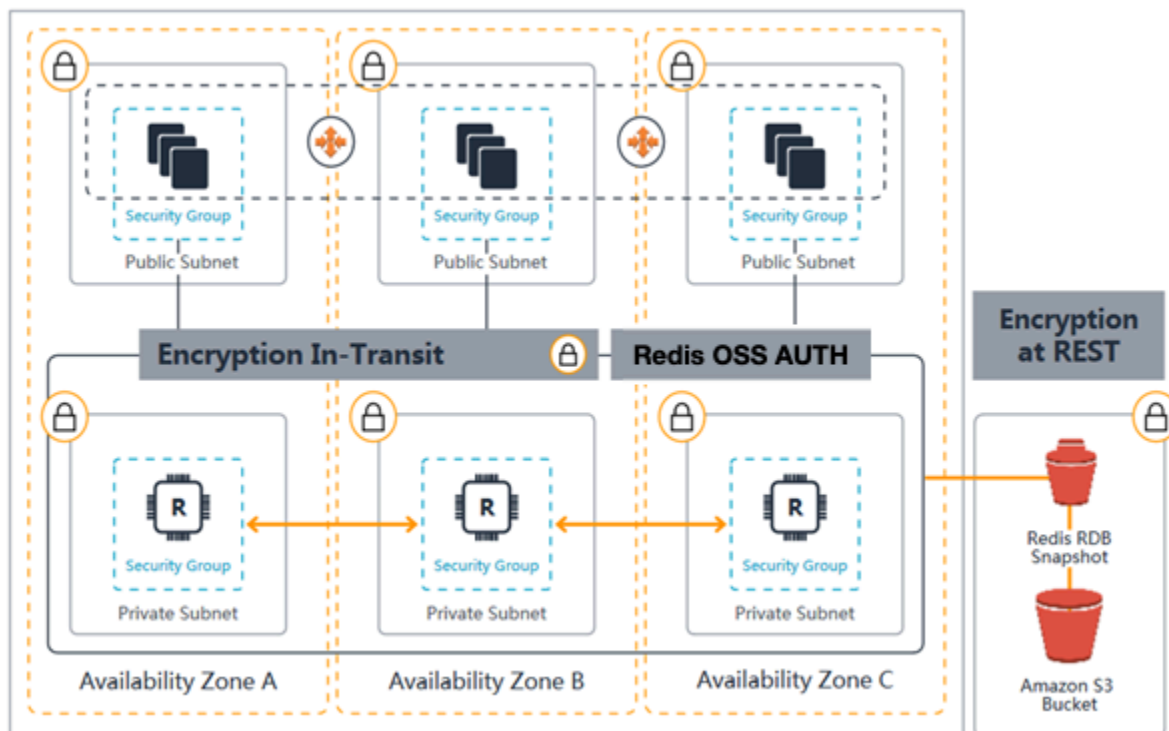
Para ajudar a manter seus dados seguros, a Amazon ElastiCache e a Amazon EC2 fornecem mecanismos de proteção contra o acesso não autorizado aos seus dados no servidor.

A Amazon ElastiCache (Memcached) fornece recursos de criptografia para dados em caches que executam as versões 1.6.12 ou posteriores do Memcached.

A Amazon ElastiCache com Valkey e Redis OSS fornece recursos de criptografia para dados em caches que executam o Valkey 7.2 ou posterior e as OSS versões 3.2.6 do Redis (programadas para, consulte o cronograma de [fim da vida útil das versões do Redis](#))EOL, [OSS 4.0.10](#) ou posteriores. A Amazon ElastiCache também oferece suporte à autenticação de usuários com Valkey e Redis OSS AUTH e à autorização de operações de usuários usando o Controle de Acesso Baseado em Funções (). IAM 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.

ElastiCache oferece suporte à autenticação de usuários usando os OSS AUTH comandos Valkey IAM e Redis e à autorização de operações de usuários usando o Controle de Acesso Baseado em Funções (). RBAC



ElastiCache com o diagrama de segurança 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 OSS sem servidor Valkey ou Redis têm a criptografia em trânsito ativada. Para clusters autoprojeto, você pode habilitar a criptografia em trânsito em um grupo de replicação definindo o parâmetro como `true` (CLI: `--transit-encryption-enabled`) `TransitEncryptionEnabled` ao criar o grupo de replicação. Você pode fazer isso se estiver criando o grupo de replicação usando o AWS Management Console AWS CLI, o ou o. ElastiCache API

Todos os caches sem servidor têm criptografia em trânsito habilitada. Para clusters autoprojeto, você pode habilitar a criptografia em trânsito em um cluster de cache definindo o parâmetro `TransitEncryptionEnabled` como `true` (CLI: `--transit-encryption-enabled`) ao criar o cluster de cache 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 de Valkey e Redis OSS](#)
- [Habilitando a criptografia em trânsito para o Memcached](#)
- [Habilitação da criptografia em trânsito](#)
- [Conectando-se à ElastiCache \(Valkey\) ou Amazon ElastiCache \(RedisOSS\) com criptografia em trânsito usando valkey-cli](#)

- [Habilitando a criptografia em trânsito em um cluster Redis OSS autoprojetoado usando Python](#)
- [Práticas recomendadas ao habilitar a criptografia em trânsito](#)
- [Conexão a nós habilitada com criptografia em trânsito usando Openssl \(Memcached\)](#)
- [Criando um cliente TLS Memcached usando Java](#)
- [Criando 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 TLS criptografadas.
- 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 OSS AUTH recurso Valkey e Redis, 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 autoprojetoada:

- A criptografia em trânsito é suportada em grupos de replicação que executam o Valkey 7.2 e posterior e as OSS versões 3.2.6, 4.0.10 e posteriores do Redis.
- A modificação da configuração de criptografia em trânsito, para um cluster existente, é suportada em grupos de replicação que executam o Valkey 7.2 e posterior e o Redis OSS versão 7 e posterior.
- A criptografia em trânsito é suportada somente para grupos de replicação executados em uma Amazon. VPC

- A criptografia em trânsito não é suportada para grupos de replicação que executam os seguintes tipos de nós: M1, M2.

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- A criptografia em trânsito é ativada configurando explicitamente o parâmetro `TransitEncryptionEnabled` como `true`.
- Certifique-se de que seu cliente de armazenamento em cache ofereça suporte à TLS conectividade e que você a tenha habilitado na configuração do cliente.
- O uso dos antigos TLS 1.0 e TLS 1.1 está obsoleto em todas as AWS regiões para a ElastiCache versão 6 e superior. ElastiCache continuará a oferecer suporte a TLS 1.0 e 1.1 até 8 de maio de 2025. Os clientes devem atualizar o software cliente antes dessa data.

### 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 autoprojetoada:

- A criptografia em trânsito é compatível com clusters executando as versões 1.6.12 e posteriores do Memcached.
- A criptografia em trânsito oferece suporte às versões 1.2 e 1.3 do Transport Layer Security (TLS).
- A criptografia em trânsito é suportada somente para clusters executados em uma AmazonVPC.
- A criptografia em trânsito não é suportada para grupos de replicação que executam os seguintes tipos de nós: 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.
- Certifique-se de que seu cliente de armazenamento em cache ofereça suporte à TLS conectividade e que você a tenha habilitado 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, você pode reduzir o impacto no desempenho da criptografia em trânsito persistindo suas TLS conexões.

### Outras opções de Valkey e Redis OSS

Para obter mais informações sobre as opções disponíveis para Valkey e RedisOSS, consulte os links a seguir.

- [Criptografia em repouso em ElastiCache](#)
- [Autenticação com o comando Valkey e Redis OSS AUTH](#)
- [Controle de acesso baseado em funções \(\) RBAC](#)
- [Amazon VPCs e ElastiCache segurança](#)
- [Identity and Access Management para Amazon ElastiCache](#)

### Habilitando a 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 para Valkey ou Redis OSS](#).

### Habilitação da criptografia em trânsito

Todos os caches sem servidor têm criptografia em trânsito habilitada. Em um cluster autoprotetido, você pode habilitar a criptografia em trânsito usando o AWS Management Console AWS CLI, o ou o ElastiCache API

## Habilitando a criptografia em trânsito usando o AWS Management Console

### Habilitando a criptografia em trânsito para um novo cluster autoprojetoado usando o AWS Management Console

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 Valkey \(modo de cluster desativado\) \(console\)](#)
- [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

### Habilitando a criptografia em trânsito para um cluster autoprojetoado existente usando o AWS Management Console

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 OSS clientes Valkey ou Redis se conectem usando conexões criptografadas e não criptografadas. Depois de migrar todos os seus OSS clientes Valkey ou Redis para usar conexões criptografadas, você poderá modificar a configuração do cluster para definir o modo de criptografia de 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 Transit como Preferencial

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Escolha caches Valkey ou OSScaches Redis 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 OSS clientes Valkey ou Redis para usar conexões criptografadas:

Defina o modo de criptografia Transit como Obrigatório

1. Faça login no AWS Management Console e abra o ElastiCache console da Amazon em <https://console.aws.amazon.com/elasticache/>.
2. Escolha caches Valkey ou OSScaches Redis 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 OSS replicação Valkey ou Redis usando o AWS CLI, use o parâmetro `transit-encryption-enabled`

Habilitando a criptografia em trânsito em um novo cluster autoprojeto para Valkey ou Redis OSS (modo de cluster desativado) () CLI

Use a AWS CLI operação `create-replication-group` e os parâmetros a seguir para criar um grupo de OSS replicação Valkey ou Redis com réplicas que tenham a criptografia em trânsito ativada:

Principais parâmetros:

- **--engine**—Deve ser `valkey` ou `redis`.
- **--engine-version**—Se o motor for RedisOSS, deverá ser 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 obter mais informações, consulte as informações a seguir.

- [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) do zero \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitando a criptografia em trânsito em um novo cluster autoprojeto para Valkey ou Redis OSS (modo de cluster ativado) () 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**—Deve ser `valkey` ou `redis`.
- **--engine-version**—Se o motor for RedisOSS, deverá ser 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 obter mais informações, consulte as informações a seguir.

- [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster ativado\) do zero \(\)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 OSS clientes Valkey ou Redis se conectem usando conexões criptografadas e não criptografadas. Depois de migrar todos os seus OSS clientes Valkey ou Redis para usar conexões criptografadas, você poderá modificar a configuração do cluster para definir o modo de criptografia de 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`.

Conectando-se à ElastiCache (Valkey) ou Amazon ElastiCache (RedisOSS) com criptografia em trânsito usando `valkey-cli`

Para acessar dados de caches ElastiCache (RedisOSS) habilitados com criptografia em trânsito, você usa clientes que trabalham com Secure Socket Layer (SSL). Você também pode usar `valkey-cli` com TLS no Amazon Linux e no Amazon Linux 2. Se seu cliente não oferecer suporte TLS, você poderá usar o `stunnel` comando no host do cliente para criar um SSL túnel para os OSS nós do Redis.

## Conexão criptografada com Linux

Para usar o valkey-cli para se conectar a um OSS cluster Valkey ou Redis habilitado com criptografia em trânsito no Amazon Linux 2023, 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 EC2 instância, digite os comandos apropriados para a versão do Linux que você está usando.

### Amazon Linux 2023

Se estiver usando o Amazon Linux 2023, digite o seguinte:

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

Em seguida, digite o comando a seguir, substituindo o endpoint do cluster e da porta pelo que é mostrado neste exemplo.

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

Para obter mais informações sobre como localizar o endpoint, consulte [Localize seus endpoints de nó](#).

### 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/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
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`, é recomendável executar o comando opcional `make-test`
4. Para se conectar a um cluster com criptografia e autenticação ativadas, 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 `redis6-cli` o comando em vez de: `valkey-cli`

```
redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
```

## Conexão criptografada com stunnel

Para usar `valkey-cli` para se conectar a um OSS cluster Redis habilitado com criptografia em trânsito usando `stunnel`, siga estas etapas.

1. Use SSH para se conectar ao seu cliente e instalar `stunnel`.

```
sudo yum install stunnel
```

2. Execute o comando a seguir para criar e editar arquivos `/etc/stunnel/valkey-cli.conf` simultaneamente para adicionar um endpoint de cluster ElastiCache (RedisOSS) 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 como o OSS padrão `127.0.0.1` do Redis 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 para o endpoint do OSS servidor Redis. Para obter mais informações, consulte [Encontrando pontos de extremidade de conexão em ElastiCache](#).

3. Inicie `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 OSS nó Redis criptografado usando o ponto final local do túnel.

- Se nenhuma AUTH senha foi usada durante a criação do cluster ElastiCache (RedisOSS), este exemplo usa o valkey-cli para se conectar ao servidor ElastiCache (RedisOSS) usando o caminho completo para valkey-cli, no Amazon Linux:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
```

Se a AUTH senha foi usada durante a criação do OSS cluster Redis, este exemplo usa valkey-cli para se conectar ao OSS servidor Redis usando o caminho completo para 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 AUTH senha foi usada durante a criação do cluster ElastiCache (RedisOSS), este exemplo usa o valkey-cli para se conectar ao servidor ElastiCache (RedisOSS) usando o caminho completo para valkey-cli, no Amazon Linux:

```
src/valkey-cli -h localhost -p 6379
```

Se a AUTH senha foi usada durante a criação do OSS cluster Redis, este exemplo usa valkey-cli para se conectar ao OSS servidor Valkey ou Redis usando o caminho completo para valkey-cli, no Amazon Linux:

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

Este exemplo usa Telnet para se conectar ao servidor Valkey RedisOSS.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Para parar e fechar os SSL túneis, o processo de `pkill` atordoamento.

```
sudo pkill stunnel
```

Habilitando a criptografia em trânsito em um cluster Redis OSS autoprojeto usando Python

O guia a seguir demonstrará como habilitar a criptografia em trânsito em um cluster Redis OSS 7.0 que foi originalmente criado com a criptografia em trânsito desativada. TCPe TLS os clientes 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 boto3 SDK para operações de ElastiCache gerenciamento (criação de cluster ou usuário) e o `redis-py-cluster redis-py/` para tratamento de dados.
- Você deve usar pelo menos a versão boto3 (=~) 1.26.39 para usar a TLS migração on-line com a modificação do cluster. API



- ElastiCache oferece suporte à TLS migração 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 OSS versão do Redis anterior à 7.0, precisará atualizar a OSS versão Redis do seu cluster. Para obter mais informações sobre as diferenças entre as versões, consulte [Principais diferenças de comportamento e compatibilidade da versão com o Redis OSS](#).

## Tópicos

- [Defina as constantes de string que iniciarão o ElastiCache Valkey ou o Redis Cluster OSS](#)
- [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 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 Cluster OSS

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 uma configuração de um cluster, que conterá metadados sobre o cluster, como a OSS versão Valkey ou Redis, o tipo de instância e se a criptografia em trânsito () TLS está ativada ou desativada.

## #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 ElastiCache Valkey ou o próprio cluster RedisOSS. Essa classe terá um campo de cliente que conterá um cliente boto3 para operações ElastiCache de gerenciamento, como criar o cluster e consultar o. ElastiCache API

```

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 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 `redis-py-cluster` cliente que se conectará ao cluster com uma TCP conexão clara semTLS.
3. O cliente `redis-py-cluster` preenche o cluster com alguns dados.
4. O cliente boto3 acionará a TLS migração de não TLS para TLS preferencial.
5. Enquanto o cluster estiver sendo migrado para TLSPreferred, o `redis-py-cluster` TCP cliente enviará get operações repetidas para o cluster até que a migração seja concluída.
6. Depois que a migração para TLS Preferred for concluída, afirmaremos que o cluster oferece suporte à criptografia em trânsito. Depois, criaremos um `redis-py-cluster` cliente que se conectará ao cluster comTLS.
7. Enviaremos alguns get comandos usando o novo TLS cliente e o TCP cliente antigo.
8. O cliente boto3 acionará a TLS migração do TLS Preferred para o necessárioTLS.
9. Enquanto o cluster está sendo migrado para o modo TLS necessário, o `redis-py-cluster` TLS cliente 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 um tratamento adequado DNS dos registros

### Note

Estamos alterando e excluindo endpoints antigos durante esse processo. O uso incorreto dos endpoints pode fazer com que o OSS cliente Valkey ou Redis 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, os registros antigos por nó são mantidos e os novos DNS registros por nó DNS são gerados em um formato diferente. TLSos clusters habilitados usam um formato de DNS registro diferente dos non-TLS-enabled clusters. ElastiCache manterá os dois DNS registros quando um cluster for configurado no modo de criptografia: preferencial para que aplicativos e outros OSS clientes Valkey ou Redis possam alternar entre eles. As seguintes alterações nos DNS registros ocorrem durante o processo TLS de migração:

Descrição das alterações nos DNS registros que ocorrem ao ativar a criptografia em trânsito

Para CME clusters

Quando um cluster é definido como “modo de criptografia em trânsito: preferencial”:

- Os endpoints originais do cluster para clusters não TLS habilitados permanecerão ativos. Não haverá tempo de inatividade quando o cluster for reconfigurado do modo de TLS criptografia 'nenhum' para 'preferido'.
- Novos OSS endpoints TLS Valkey ou Redis serão gerados quando o cluster for definido no modo -preferencial. TLS Esses novos endpoints serão resolvidos da IPs mesma forma que os antigos (não-TLS).
- O novo endpoint de OSS configuração TLS Valkey ou Redis será exposto no ElastiCache console e na resposta a. `describe-replication-group` API

Quando um cluster é definido como “modo de criptografia em trânsito: obrigatório”:

- Os endpoints antigos não TLS habilitados serão excluídos. Não haverá tempo de inatividade dos endpoints do TLS cluster.

- Você pode recuperar um novo `cluster-configuration-endpoint` do ElastiCache Console ou do `describe-replication-group` API.

Para CMD clusters 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 do cluster não TLS habilitado permanecerão ativos.
- Novos endpoints TLS primários e de leitura serão gerados quando o cluster for configurado para TLS Preferred modo. Esses novos endpoints serão resolvidos para o (s) mesmo (s) IP (s) dos antigos (não-TLS).
- O novo endpoint primário e o endpoint do leitor serão expostos no ElastiCache console e na resposta ao `describe-replication-group` API

Quando o grupo de replicação é definido como “modo de criptografia em trânsito: obrigatório”:

- Os endpoints antigos não TLS primários e de leitura serão excluídos. Não haverá tempo de inatividade dos endpoints do TLS cluster.
- Você pode recuperar novos endpoints primários e de leitura do ElastiCache Console ou do `describe-replication-group` API

O uso sugerido dos DNS registros

Para CME clusters

- Use o endpoint de configuração do cluster em vez dos DNS registros por nó no código do seu aplicativo. Não é recomendável usar DNS nomes por nó diretamente, pois eles podem mudar ao adicionar ou remover fragmentos.
- Não codifique o 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 o `describe-replication-group` API (conforme demonstrado acima (em negrito)) e use a DNS resposta obtida a partir desse momento.

## Para CMD clusters com o failover automático ativado

- Use o endpoint primário e o endpoint do leitor em vez dos DNS nomes por nó no código do seu aplicativo, pois os DNS nomes antigos por nó são excluídos e os novos são gerados ao migrar o cluster de não para -preferencial. TLS Não é recomendável usar DNS nomes 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 ElastiCache serviço. O uso do endpoint primário e do endpoint do leitor é sugerido 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 eles podem ser alterados durante o processo de migração. TLS Depois que a alteração da migração para TLS -preferred for concluída, consulte o endpoint primário e o endpoint do leitor com o describe-replication-group API e use a resposta DNS que você obtiver a partir desse momento. Dessa forma, você poderá acompanhar as mudanças nos endpoints de forma dinâmica.

## Para CMD cluster com o failover automático desativado

- Use o endpoint primário e o endpoint do leitor em vez dos DNS nomes por nó no código do seu aplicativo. Quando o Failover Automático está desativado, o escalonamento, a correção, o failover e outros procedimentos que são gerenciados automaticamente pelo ElastiCache serviço quando o Failover Automático está ativado são feitos por você. Isso facilita seu acompanhamento manual dos diferentes endpoints. Como os DNS nomes antigos por nó são excluídos e novos são gerados ao migrar o cluster de não TLS para TLS -preferencial, não use os nomes por DNS nó diretamente. Isso é obrigatório para que os clientes possam se conectar ao cluster durante a TLS migração. Além disso, você se beneficiará de distribuir uniformemente as leituras entre as réplicas ao usar o endpoint do leitor e acompanhar os DNS -records 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 TLS processo de migração.

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 termina a migração para TLS -preferred, ele é capaz de aceitar e atender ambas as conexõesTCP. TLS Portanto, você não deve criar clientes que tentarão estabelecer TLS conexões 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):

- Usando o SNS serviço para receber uma notificação quando a criptografia for concluída
- Usar o `describe-events` API que emitirá um evento quando a criptografia for concluída
- Vendo uma mensagem no ElastiCache console 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”)
- Afirmando que o cluster foi `transit_encryption_enabled` definido como `True` consultando o `describe-replication-group` API

Depois de ativar a criptografia em trânsito: verifique se os clientes que você usa estão configurados corretamente

Enquanto o cluster estiver no modo TLS preferencial, seu aplicativo deve abrir TLS conexões 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 TCP conexões mais claras com o OSS mecanismo Valkey ou Redis 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
```

Conexão a nós habilitada com criptografia em trânsito usando Openssl (Memcached)

Para acessar dados de nós 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
```

## Criando um cliente TLS Memcached usando Java

Para criar um cliente no TLS modo, faça o seguinte para inicializar o cliente com o `apropradoSSLContext`:

```
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");
    }
}
```

## Criando um cliente TLS Memcached usando PHP

Para criar um cliente no TLS modo, faça o seguinte para inicializar o cliente com o `apropradoSSLContext`:

```
<?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->getLastErrorMessage(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
* See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
* Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */
```

```
/* store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
if($tls_client->set('key', 'value', 60)) {
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
} else {
    echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
```

Para obter mais informações sobre como usar o PHP cliente, 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 ter 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 AWS KMS chaves 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](#).

### Note

A criptografia padrão (gerenciada pelo serviço) é a única opção disponível nas regiões GovCloud (EUA).

### Important

Habilitar a criptografia em repouso em um OSS cluster Valkey ou Redis autoprojeto existente envolve a exclusão do grupo de replicação existente, depois de executar o 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 de 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 suportada em grupos de replicação que executam o Valkey 7.2 e versões posteriores e OSS as versões do Redis (3.2.6 programadas para, consulte o cronograma de [fim de vida das OSS versões do Redis](#))EOL, 4.0.10 ou posteriores.
- A criptografia em repouso é suportada somente para grupos de replicação executados em uma Amazon. VPC
- A criptografia em repouso é compatível somente com grupos de replicação que estejam executando os tipos de nó a seguir.
  - R6gd, R6g, R5, R4, R3
  - M6g, M5, M4, M3
  - T4g, T3, T2

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 serão criptografados, independentemente de a criptografia em SSD 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 serão criptografados com a AWS KMS chave SSD gerenciada pelo cliente escolhida (ou criptografia gerenciada pelo serviço nas regiões). AWS GovCloud
- Com o Memcached, a criptografia em repouso é suportada somente em caches 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 de AWS KMS

ElastiCache suporta AWS KMS chaves simétricas gerenciadas pelo cliente (KMSchave) para criptografia em repouso. KMSAs chaves gerenciadas pelo cliente são chaves de criptografia que você cria, possui e gerencia em sua AWS conta. Para obter mais informações, consulte [AWS KMSas chaves](#) no Guia do desenvolvedor do AWS Key Management Service. As chaves devem ser criadas AWS KMS antes que possam ser usadas com ElastiCache.

Para saber como criar chaves AWS KMS raiz, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

ElastiCache permite que você se integre com 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 AWS KMS.

A chave de `kms:ViaService` condição limita o uso de uma AWS KMS chave (KMSchave) às solicitações de AWS serviços especificados. 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 API chamadas AWS Key Management Service

relacionadas às chaves gerenciadas pelo cliente têm CloudTrail registros correspondentes. Você também pode ver as concessões ElastiCache criadas ao ligar para a [ListGrants](#)KMSAPI chamada.

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 KMS chave 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 em um novo objeto do S3 e criptografar usando uma KMS chave gerenciada pelo cliente, copiar o arquivo para outro bucket do S3 configurado com criptografia padrão usando uma KMS chave 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 KMS chave, mesmo que os dados não estejam 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. 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 de suas chaves AWS KMS raiz, 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 [AWS KMS Chaves rotativas](#) no Guia do desenvolvedor do AWS Key Management Service.
- Criptografar um ElastiCache cache usando uma KMS chave 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 AWS KMS concessões e limites, 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 autoprojeto, 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 o AWS Management Console, o AWS CLI, ou o 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 — Esta opção permite que você forneça o ID/formulário ARN da chave AWS KMS para criptografia em repouso.

Para saber como criar chaves AWS KMS raiz, consulte [Criar chaves](#) no Guia do desenvolvedor do AWS Key Management Service

## Sumário

- [Habilitando a criptografia em repouso usando o AWS Management Console](#)
- [Habilitando a criptografia em repouso usando o AWS CLI](#)

Habilitando a criptografia em repouso em um cluster Valkey ou Redis autoprojeto existente OSS

Você só pode ativar a criptografia em repouso ao criar um grupo de replicação Valkey ou RedisOSS. 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 obter mais informações, consulte [Excluindo um cluster no ElastiCache](#) ou [Exclusão de um grupo de replicação](#).

Habilitando a criptografia em repouso usando o AWS Management Console

Habilitação da criptografia em repouso em um cache sem servidor (console)

Todos os caches sem servidor têm criptografia em repouso habilitada. Por padrão, uma KMS chave AWS 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 Cliente gerenciado CMK em Configuração de chave de criptografia.

- Selecione uma chave na configuração Chave AWS KMS .

Habilitação da criptografia em repouso em um cluster autoprojeto 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 chave gerenciada pelo cliente CMK.

Para o step-by-step procedimento, consulte o seguinte:

- [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)
- [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

Habilitando a criptografia em repouso usando o AWS CLI

Para habilitar a criptografia em repouso ao criar um OSS cluster Valkey ou Redis usando o AWS CLI, use o `at-rest-encryption-enabled` parâmetro -- ao criar um grupo de replicação.

Habilitando a criptografia em repouso em um cluster Valkey ou Redis OSS (modo de cluster desativado) () 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`—Deve ser `valkey` ou `redis`.
- `--engine-version`—Se o motor for RedisOSS, deverá ser 3.2.6, 4.0.10 ou posterior.
- `--at-rest-encryption-enabled`: obrigatório para habilitar a criptografia em repouso.

## Example 1: Cluster Valkey ou Redis OSS (modo de cluster desativado) com réplicas

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-classic-rg \  
  --replication-group-description "3 node replication group" \  
  --cache-node-type cache.m4.large \  
  --engine redis \  
  --at-rest-encryption-enabled \  
  --num-cache-clusters 3
```

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:

- [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) do zero \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitando a criptografia em repouso em um cluster para Valkey ou Redis OSS (modo de cluster ativado) () 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**—Deve ser `valkey` ou `redis`.
- **--engine-version**—Se o motor for RedisOSS, deverá ser 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 Valkey ou Redis OSS (modo de cluster ativado)

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:

- [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster ativado\) do zero \(AWS CLI\)](#)

- [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 web que ajuda você a controlar com segurança o acesso aos AWS recursos. ElastiCache oferece suporte à autenticação de usuários usando os OSS AUTH comandos Valkey IAM e Redis e à autorização de operações de usuários usando o Controle de Acesso Baseado em Funções (). RBAC

Tópicos

- [Controle de acesso baseado em funções \(\) RBAC](#)
- [Autenticação com o comando Valkey e Redis OSS AUTH](#)
- [Desabilitando o controle de acesso em um cache ElastiCache Valkey ou Redis OSS](#)

### Controle de acesso baseado em funções () RBAC

Em vez de autenticar usuários com o OSS AUTH comando Valkey e Redis, conforme descrito em [Autenticação com o comando Valkey e Redis OSS AUTH](#), no Valkey 7.2 e versões posteriores e no Redis OSS 6.0 em diante, você pode usar um recurso chamado Controle de Acesso Baseado em Função (). RBAC RBAC também é a única maneira de controlar o acesso aos caches sem servidor.

Ao contrário do Valkey e do Redis OSS AUTH, em que todos os clientes autenticados têm acesso total ao cache se o token for autenticado, 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 RBAC, você cria usuários e atribui a eles permissões específicas usando uma string de acesso, conforme descrito a seguir. Você atribui os usuários a grupos de usuários alinhados com uma função específica (administradores, recursos humanos) que são então implantados em um ou mais caches. ElastiCache Ao fazer isso, você pode estabelecer limites de segurança entre clientes usando o mesmo OSS cache ou caches Valkey ou Redis e impedir que os clientes acessem os dados uns dos outros.

### Note

Ao usar RBAC com clusters Valkey, você ainda precisará atribuir aos usuários e grupos de usuários o mecanismo “redis”.

RBAC foi projetado para oferecer suporte à introdução do [ACL](#) no Redis OSS 6. Quando você usa RBAC com seu OSS cache ElastiCache Valkey ou Redis, há algumas limitações:

- Não é possível especificar senhas em uma string de acesso. Você define senhas com [CreateUser](#) e [ModifyUser](#) chamadas.
- Para direitos de usuário, você passa 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.
- Você não pode usar comandos proibidos e renomeados. 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 API parâmetros e ElastiCache (RedisOSS) gerencia senhas. Assim, você não pode usar `reset` porque ele removeria todas as senhas de um usuário.
- O Redis OSS 6 apresenta o [ACLLIST](#) comando. Esse comando retorna uma lista de usuários junto com as ACL regras aplicadas a cada usuário. ElastiCache (RedisOSS) suporta o ACL LIST comando, mas não inclui suporte para hashes de senha, como faz o RedisOSS. Com o ElastiCache (RedisOSS), você pode usar a operação [describe-users](#) para obter informações semelhantes, incluindo as regras contidas na string de acesso. No entanto, [describe-users](#) não recupera uma senha de usuário.

Outros comandos somente para leitura compatíveis ElastiCache com Valkey e Redis OSS incluem [ACLWHOAMI](#), e [ACLUSERSACLCAT](#). ElastiCache com Valkey e Redis OSS não oferece suporte a nenhum outro comando baseado em gravação ACL.

- As limitações a seguir aplicam-se:

Recurso	Máximo permitido
Usuários por grupo de usuários	100

Recurso	Máximo permitido
Número de DPUs	1000
Número de grupos de usuários	100

O uso RBAC com ElastiCache (RedisOSS) é descrito em mais detalhes a seguir.

## Tópicos

- [Especificação de permissões usando uma string de acesso](#)
- [Aplicando RBAC a um cache ElastiCache com Valkey ou Redis OSS](#)
- [Migrando de para AUTH RBAC](#)
- [Migrando de para RBAC AUTH](#)
- [Alternância automática de senhas para os usuários](#)
- [Autenticação com IAM](#)

## Especificação de permissões usando uma string de acesso

Para especificar permissões para um cache ElastiCache (RedisOSS), você cria uma string de acesso e a atribui a um usuário, usando o AWS CLI ou AWS Management Console.

As strings 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 ACL regras, 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 *comando1*.

`+@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 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 a OSS versão 6.2 e superior do Redis, a seguinte sintaxe de string de acesso também é suportada:

- `&*`: o acesso é dado a todos os canais disponíveis.

Para a OSS versão 7.0 e superior do Redis, a seguinte sintaxe de string de acesso também é suportada:

- `|`: 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. Consulte [as ACL principais permissões](#) para obter mais informações.

- `%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. Veja mais informações sobre [ACLos seletores](#).
- `clearselectors`: exclua todos os seletores associados ao usuário.

## Aplicando RBAC a um cache ElastiCache com Valkey ou Redis OSS

Para usar ElastiCache com o Valkey ou o Redis OSSRBAC, 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 CLI](#)
- [Gerenciando grupos de usuários com o console e 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 CLI

As informações do usuário para RBAC usuários são uma 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.



```
--passwords "a-strong-pa))word" \  
--access-string "off +get ~keys*"
```

Para Windows:

```
aws elasticache create-user ^  
--user-id "new-default-user" ^  
--user-name "default" ^  
--engine "REDIS" ^  
--passwords "a-strong-pa))word" ^  
--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 "REDIS" \  
--user-ids "new-default-user"
```

Para Windows:

```
aws elasticache create-user-group ^  
--user-group-id "new-group-2" ^  
--engine "REDIS" ^  
--user-ids "new-default-user"
```

3. Troque o novo usuário default pelo usuário original default.

Para Linux, macOS ou Unix:

```
aws elasticache modify-user-group \  
--user-group-id test-group \  
--user-ids-to-add "new-default-user" \  
--user-ids-to-remove "default"
```

Para Windows:

```
aws elasticache modify-user-group ^  
--user-group-id test-group ^
```



```
--user-ids-to-add "new-default-user" ^  
--user-ids-to-remove "default"
```

Quando essa operação de modificação é chamada, todas as conexões existentes com um cache que o usuário padrão original tenha são encerradas.

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 RBAC for ElastiCache (RedisOSS):

- As senhas devem ter de 16 a 128 caracteres imprimíveis.
- Os seguintes caracteres não alfanuméricos não são permitidos: , " " / @.

### Gerenciando usuários com o console e CLI

Use o procedimento a seguir para gerenciar usuários no console.

Para gerenciar usuários no console

1. Faça login no AWS Management Console 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 o 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:*" \  
  --no-password-required
```

Para Windows:

```
aws elasticache modify-user ^  
  --user-id user-id-1 ^  
  --access-string "~objects:* ~items:* ~public:*" ^  
  --no-password-required
```

#### 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 o 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 a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache delete-user \  
  --user-id user-id-1
```

```
--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
```

### Gerenciando grupos de usuários com o console e 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 AWS Management Console 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


Ao criar um grupo de usuários, você precisa incluir o usuário padrão.

- 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 usando CLI o.

Para criar um novo grupo de usuários e adicionar um usuário usando o 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 "REDIS" \  
  --user-ids user-id-1, user-id-2
```

Para Windows:

```
aws elasticache create-user-group ^  
  --user-group-id "new-group-1" ^  
  --engine "REDIS" ^  
  --user-ids user-id-1, user-id-2
```

Para modificar um grupo de usuários adicionando novos usuários ou removendo membros atuais usando o 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 userid-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 o 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 de criar um grupo de usuários e adicionar usuários, a etapa final da implementação RBAC é 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 AWS Management Console, faça o seguinte:

- Para o modo cluster desabilitado, consulte [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)
- Para o modo cluster habilitado, consulte [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

## 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**— Deve 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 "redis" \  
  --user-group-id "new-group-1"
```

Para Windows:

```
aws elasticache create-serverless-cache ^
  --serverless-cache-name "new-serverless-cache" ^
  --description "new-serverless-cache" ^
  --engine "redis" ^
  --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 da implementação 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 AWS Management Console, faça o seguinte:

- Para o modo cluster desabilitado, consulte [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)
- Para o modo cluster habilitado, consulte [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#)

## 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**— Deve 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 "redis" \  
  --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 "redis" ^  
  --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](#).

### Migrando de para AUTH RBAC

Se você estiver usando AUTH conforme descrito em [Autenticação com o comando Valkey e Redis OSS AUTH](#) e quiser migrar para o uso RBAC, use os procedimentos a seguir.

Use o procedimento a seguir para migrar do console AUTH para o RBAC uso.

Para migrar do Valkey ou do Redis OSS AUTH para usar o console RBAC

1. Faça login no AWS Management Console 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 Valkey ou do Redis OSS AUTH para RBAC o uso do CLI

Para migrar de AUTH para o RBAC uso do 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
```

Migrando de para RBAC AUTH

Se você estiver usando RBAC e quiser migrar para o Redis OSSAUTH, consulte. [Migrando de para RBAC 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 “Desabilitando o controle de acesso em um cache 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 API chamada 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 senhas ElastiCache (RedisOSS) (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 tem um conjunto de recursos equivalente ao Redis OSS 7.0. No Redis OSS 6, ElastiCache introduziu [Controle de acesso baseado em funções \(\) RBAC](#) para proteger o cluster Valkey ou RedisOSS. Esse recurso permite que certas conexões sejam limitadas em termos dos comandos que podem ser executados e das chaves que podem ser acessadas. Com RBAC, enquanto o cliente cria um usuário com senhas, os valores da senha precisam ser inseridos manualmente em texto simples 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 ElastiCache lateral.

```
{
  "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 ativar a rotação automática de senhas do Secrets Manager, você criará uma função Lambda que interagirá com o usuário [modificado API 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 `aws:SourceArn` condição em sua política de função de rotação, a função de rotação só poderá ser usada para alternar o segredo especificado por elaARN. 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 AWS Management Console 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 ElastiCache usuário do usuário,
  - iv. Escolha Salvar.
- g. Escolha Permissions (Permissões)
- h. Em Função de execução, escolha o nome da função Lambda para visualizar no IAM console.
- i. A função do Lambda precisará da seguinte permissão para modificar os usuários e definir a senha:

ElastiCache

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:DescribeUsers",
      "elasticache:ModifyUser"
    ],
    "Resource": "arn:aws:elasticache:us-east-1:xxxxxxxxxxx918:user:user1"
  }
]
}

```

## Secrets Manager

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:UpdateSecretVersionStage"
      ],
      "Resource": "arn:aws:secretsmanager:us-east-1:xxxxxxxxxxx:secret:XXXX"
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetRandomPassword",
      "Resource": "*"
    }
  ]
}

```

### 4. Configurar a alternância de segredos do Secrets Manager

- a. Usando o AWS Management Console, 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 IAM

### Tópicos

- [Visão geral](#)
- [Limitações](#)
- [Configuração](#)
- [Conexão](#)

### Visão geral

Com a IAM Autenticação, você pode autenticar uma conexão ElastiCache com Valkey ou Redis OSS usando AWS IAM identidades, quando seu cache estiver configurado para usar Valkey ou OSS Redis 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 IAM Autenticação para configurar um controle de acesso refinado para cada ElastiCache cache e ElastiCache usuário individual, seguindo os princípios de permissões de privilégio mínimo. IAMA autenticação ElastiCache com Valkey ou Redis OSS funciona fornecendo um token de IAM autenticação de curta duração em vez de uma senha de ElastiCache usuário de longa duração no comando Valkey ou Redis ou. OSS AUTH HELLO Para obter mais informações sobre o token de IAM autenticação, consulte o [processo de assinatura do Signature Versão 4](#) no Guia de Referência AWS Geral e no exemplo de código abaixo.

Você pode usar IAM identidades e suas políticas associadas para restringir ainda mais o acesso ao Valkey ou ao RedisOSS. Você também pode conceder acesso aos usuários de seus provedores de identidade federados diretamente aos caches Valkey ou RedisOSS.

Para usar 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 IAM identidade. A IAM identidade 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 IAM autenticação usando AWS as credenciais do IAM



usuário ou da função. Finalmente, você precisa fornecer o token de IAM autenticação de curta duração como uma senha em seu Valkey ou Redis OSS Client ao se conectar ao seu cache. Um OSS cliente Valkey ou Redis com suporte para provedor de credenciais pode gerar automaticamente as credenciais temporárias para cada nova conexão. ElastiCache realizará a IAM autenticação para solicitações de conexão de ElastiCache usuários IAM habilitados e validará as solicitações de conexão com. IAM

## Limitações

Ao usar a IAM autenticação, as seguintes limitações se aplicam:

- IAMa autenticação está disponível ao ser usada ElastiCache com o Valkey 7.2 ou superior e o Redis OSS versão 7.0 ou superior.
- Para ElastiCache usuários IAM habilitados, as propriedades de nome de usuário e ID de usuário devem ser idênticas.
- O token de IAM autenticação é válido por 15 minutos. Para conexões de longa duração, recomendamos usar um OSS cliente Valkey ou Redis que ofereça suporte a uma interface de provedor de credenciais.
- Uma conexão IAM autenticada ElastiCache com Valkey ou Redis OSS será automaticamente desconectada após 12 horas. A conexão pode ser prolongada por 12 horas enviando um HELLO comando AUTH or com um novo token de IAM autenticação.
- IAMa autenticação não é suportada nos MULTI EXEC comandos.
- Atualmente, a IAM autenticação suporta as seguintes chaves de contexto de condição global:
  - Ao usar a IAM autenticação com caches sem servidor,,,aws:VpcSourceIp, aws:SourceVpc, aws:SourceVpcId, aws:CurrentTime,aws:EpochTime, e aws:ResourceTag/%s (de caches e usuários sem servidor associados) são suportados.
  - Ao usar a IAM autenticação com grupos de replicação, aws:SourceIp e aws:ResourceTag/%s (de grupos de replicação e usuários associados) são suportados.

Para obter mais informações sobre chaves de contexto de condição [AWS global, consulte chaves de contexto de condição global](#) no Guia IAM do usuário.

## Configuração

Para configurar a IAM autenticação:

### 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 IAM confiança, conforme mostrado abaixo, para sua função, que permita que sua conta assuma a nova função. Salve a política em um arquivo chamado `trust-policy.json`.

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Allow",  
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
    "Action": "sts:AssumeRole"  
  }  
}
```

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

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect" : "Allow",  
      "Action" : [  
        "elasticache:Connect"  
      ],  
      "Resource" : [  
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",  
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"  
      ]  
    }  
  ]  
}
```

4. Crie uma IAM função.

```
aws iam create-role \  
  --role-name "elasticache-iam-auth-app" \  
  --policy-arn "arn:aws:iam::aws:policy/AWSLambda_FullAccess"
```

```
--assume-role-policy-document file://trust-policy.json
```

## 5. Crie a IAM política.

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

## 6. Anexe a IAM política 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 IAM habilitado.

```
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, você precisa gerar o token de IAM autenticação de curta duração usando uma solicitação [AWS pré-assinada SigV4](#). Depois disso, você fornece o token de IAM autenticação como senha ao se conectar a um OSS cache Valkey ou Redis, 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
(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 IAM autenticação.

```
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 (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 OSS cliente Lettuce Redis que envolve um provedor de credenciais para IAMAuthTokenRequest gerar 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 = userId;
        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());
    }
}
```

```
}
```

## Autenticação com o comando Valkey e Redis OSS AUTH

### Note

O AUTH foi substituído por. [the section called “Controle de acesso baseado em funções \(\) RBAC”](#) Todos os caches sem servidor devem ser usados RBAC para autenticação.

Os tokens ou senhas de OSS autenticação Valkey e Redis permitem que o Valkey e o Redis OSS exijam uma senha antes de permitir que os clientes executem comandos, melhorando assim a segurança dos dados. O AUTH está disponível somente para clusters autoprojettados.

### Tópicos

- [Visão geral de AUTH in ElastiCache with Valkey e Redis OSS](#)
- [Aplicando autenticação a um cluster ElastiCache com Valkey ou Redis OSS](#)
- [Modificando o AUTH token em um cluster existente](#)
- [Migrando de para RBAC AUTH](#)

### Visão geral de AUTH in ElastiCache with Valkey e Redis OSS

Quando você usa o AUTH ElastiCache com seu OSS cluster Valkey ou Redis, há alguns refinamentos.

Em particular, esteja ciente dessas restrições de AUTH token ou senha ao usar: AUTH

- Tokens, ou senhas, devem ter de 16 a 128 caracteres imprimíveis.
- Caracteres não alfanuméricos são restritos a (!, &, #, \$, ^, <, >, -).
- AUTH só pode ser habilitado para criptografia em trânsito habilitada ElastiCache com clusters Valkey ou RedisOSS.

Para configurar um token forte, recomendamos que você siga uma política de senha estrita, tal como exigir o seguinte:

- Os tokens ou senhas devem incluir pelo menos três dos seguintes tipos de caracteres:



- Caracteres maiúsculos
- Caracteres minúsculos
- Dígitos
- Caracteres não alfanuméricos (!, &, #, \$, ^, <, >, -)
- Os tokens ou senhas não devem conter uma palavra do dicionário ou uma palavra do dicionário ligeiramente modificada.
- Os tokens ou senhas não devem ser iguais ou semelhantes a um token usado recentemente.

## Aplicando autenticação a um cluster ElastiCache com Valkey ou Redis OSS

Você pode exigir que os usuários insiram um token (senha) em um servidor Valkey ou Redis protegido por token. OSS Para fazer isso, inclua o parâmetro `--auth-token (API:AuthToken)` com o token correto ao criar seu grupo ou cluster de replicação. 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**— Deve ser `valkey` ou `redis`.
- **--engine-version**— Se o motor for RedisOSS, deverá ser 3.2.6, 4.0.10 ou posterior.
- **--transit-encryption-enabled**— Necessário para autenticação e HIPAA elegibilidade.
- **--auth-token**— Exigido para HIPAA elegibilidade. Esse valor deve ser o token correto para esse servidor Valkey ou Redis protegido por token. OSS
- **--cache-subnet-group**— Exigido para HIPAA elegibilidade.

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 \  
  --transit-encryption-enabled
```

```
--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
```

Modificando o AUTH token em um cluster existente

Para facilitar a atualização da autenticação, você pode modificar o AUTH token 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: e. ROTATE SET A ROTATE estratégia adiciona um AUTH token adicional ao servidor enquanto retém o token anterior. A SET estratégia atualiza o servidor para suportar apenas um único AUTH token. Faça essas chamadas de modificação com o parâmetro `--apply-immediately` para aplicar as alterações imediatamente.

Girando o token AUTH

Para atualizar um OSS servidor Valkey ou Redis com um novo AUTHtoken, chame o `ModifyReplicationGroup` API com o `--auth-token` parâmetro como o novo AUTH token e o `--auth-token-update-strategy` com o valor. ROTATE Depois que a ROTATE modificação for concluída, o cluster suportará o AUTH token anterior, além do especificado no `auth-token` parâmetro. Se nenhum AUTH token tiver sido configurado no grupo de replicação antes da rotação do AUTH token, o cluster suportará o AUTH token especificado no `--auth-token` parâmetro, além de oferecer suporte à conexão sem autenticação. Consulte [Configurando o AUTH token](#) para atualizar o AUTH token necessário usando a estratégia de atualização SET.

**Note**

Se você não configurar o AUTH token antes, quando a modificação for concluída, o cluster não suportará nenhum AUTH token além do especificado no parâmetro `auth-token`.

Se essa modificação for realizada em um servidor que já suporta dois AUTH tokens, o AUTH token mais antigo também será removido durante essa operação. Isso permite que um servidor ofereça suporte a até dois AUTH tokens mais recentes em um determinado momento.

Nesse ponto, você pode continuar atualizando o cliente para usar o AUTH token mais recente. Depois que os clientes forem atualizados, você poderá usar a SET estratégia de rotação de AUTH tokens (explicada na seção a seguir) para começar a usar exclusivamente 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
```

## Configurando o AUTH token

Para atualizar um OSS servidor Valkey ou Redis para oferecer suporte a um único AUTH token necessário, chame a `ModifyReplicationGroup` API operação com o `--auth-token` parâmetro com o mesmo valor do último AUTH token e o `--auth-token-update-strategy` parâmetro com o valor. SET A SET estratégia só pode ser usada com um cluster que tenha 2 AUTH tokens ou

1 AUTH token opcional do uso de uma ROTATE estratégia anterior. Depois que a modificação for concluída, o servidor suportará somente o AUTH token especificado no parâmetro auth-token.

A AWS CLI operação a seguir modifica um grupo de replicação para definir o AUTH token. *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
```

### Habilitando a autenticação em um cluster existente

Para habilitar a autenticação em um OSS servidor Valkey ou Redis existente, chame a ModifyReplicationGroup API operação. Chame ModifyReplicationGroup com o --auth-token parâmetro como o novo token e o --auth-token-update-strategy com o valor ROTATE.

Depois que a ROTATE modificação for concluída, o cluster suportará o AUTH token especificado no --auth-token parâmetro, além de oferecer suporte à conexão sem autenticação. Depois que todos os aplicativos cliente forem atualizados para se autenticar no Valkey ou no Redis OSS com o AUTH token, use a SET estratégia para marcar o AUTH token conforme necessário. A ativação da autenticação só é suportada nos OSS servidores Valkey e Redis com a criptografia em trânsito (TLS) ativada.

### Migrando de para RBAC AUTH

Se você estiver autenticando usuários com Valkey ou Redis OSS Role-Based Access Control (RBAC), conforme descrito em [Controle de acesso baseado em funções \(\) RBAC](#), e quiser migrar para, use os procedimentos a AUTH seguir. Você pode migrar usando o console ou CLI.

## Para migrar de RBAC para o AUTH uso do console

1. Faça login no AWS Management Console 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. Para Controle de acesso, escolha Acesso de usuário AUTH padrão Valkey ou acesso de usuário OSS AUTH padrão do Redis.
7. Em Token Valkey ou AUTH token Redis, defina um novo OSS AUTH token.
8. Escolha Previsualizar alterações e, na próxima tela, Modificar.

## Para migrar de RBAC para o AUTH uso do AWS CLI

Use um dos comandos a seguir para configurar um novo AUTH token opcional para seu grupo de OSS replicação Valkey ou Redis. Observe que um token de autenticação opcional permitirá acesso não autenticado ao grupo de replicação até que o token de autenticação seja marcado como obrigatório, usando a estratégia SET de atualização 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 OSS aplicativos Valkey ou Redis para se autenticar no grupo de ElastiCache replicação usando o token opcional recém-configurado. AUTH Para concluir a rotação do token Auth, use a estratégia de atualização SET no comando subsequente abaixo. Isso marcará o AUTH token opcional conforme necessário. Quando a atualização do token Auth for concluída, o status do grupo de replicação será exibido 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 [Autenticação com o comando Valkey e Redis OSS AUTH](#).

#### Note

Se você precisar desativar o controle de acesso em um ElastiCache cluster, consulte [the section called “Desabilitando o controle de acesso em um cache ElastiCache Valkey ou Redis OSS”](#).

## Desabilitando o controle de acesso em um cache ElastiCache Valkey ou Redis OSS

Siga as instruções abaixo para desativar o controle de acesso em um cache habilitado para Valkey ou Redis OSSTLS. Seu cache terá um dos dois tipos diferentes de configurações: acesso de usuário AUTH padrão ou Lista de controle de acesso de grupos de usuários (RBAC). Se seu cache foi criado com a AUTH configuração, você precisa alterá-lo para a RBAC configuração antes de poder desabilitar o cache removendo os grupos de usuários. Se seu cache foi criado com a RBAC configuração, você pode ir direto para desativá-lo.

Para desativar um cache sem OSS servidor Valkey ou Redis 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": ""
  "...
}
```

Para desativar um OSS cache Valkey ou Redis configurado com um token AUTH

1. Altere o AUTH token 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 AUTH token foi desativado 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

```

Para desativar um OSS cluster Valkey ou Redis 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

A Amazon ElastiCache usa as seguintes técnicas para proteger seus dados de cache e protegê-los contra 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 VPC endpoints de interface \(AWS PrivateLink\)](#)
- [Sub-redes e grupos de sub-redes](#)

## Amazon VPCs e ElastiCache segurança

Como a segurança dos dados é importante, 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 (AmazonVPC) 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 Amazon Virtual Private Cloud (AmazonVPC) define uma rede virtual que se assemelha muito a um data center tradicional. Ao configurar sua Amazon, VPC você pode selecionar seu intervalo de endereços IP, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações de segurança. Você também pode adicionar um cluster de cache à rede virtual e controlar o acesso ao cluster de cache usando grupos de VPC segurança da Amazon.

Esta seção explica como configurar manualmente um ElastiCache cluster em uma AmazonVPC. Essas informações são destinadas a usuários que desejam uma compreensão mais profunda de como ElastiCache a Amazon VPC trabalha em conjunto.

### Tópicos

- [Compreensão ElastiCache e Amazon VPCs](#)
- [Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC](#)
- [Criando uma nuvem privada virtual \(VPC\)](#)
- [Conectando-se a um cache em execução em uma Amazon VPC](#)

## Compreensão ElastiCache e Amazon VPCs

ElastiCache está totalmente integrado com a Amazon Virtual Private Cloud (AmazonVPC). Para ElastiCache os usuários, isso significa o seguinte:

- Se sua AWS conta suportar apenas a VPC plataforma EC2 -, ElastiCache sempre inicie seu cluster em uma AmazonVPC.
- Se você é novato AWS, seus clusters serão implantados em uma AmazonVPC. Um padrão VPC será criado automaticamente para você.
- Se você tem um padrão VPC e não especifica uma sub-rede ao iniciar um cluster, o cluster é iniciado na sua Amazon VPC padrão.

Para obter mais informações, consulte [Detectando suas plataformas suportadas e se você tem um padrão VPC](#).

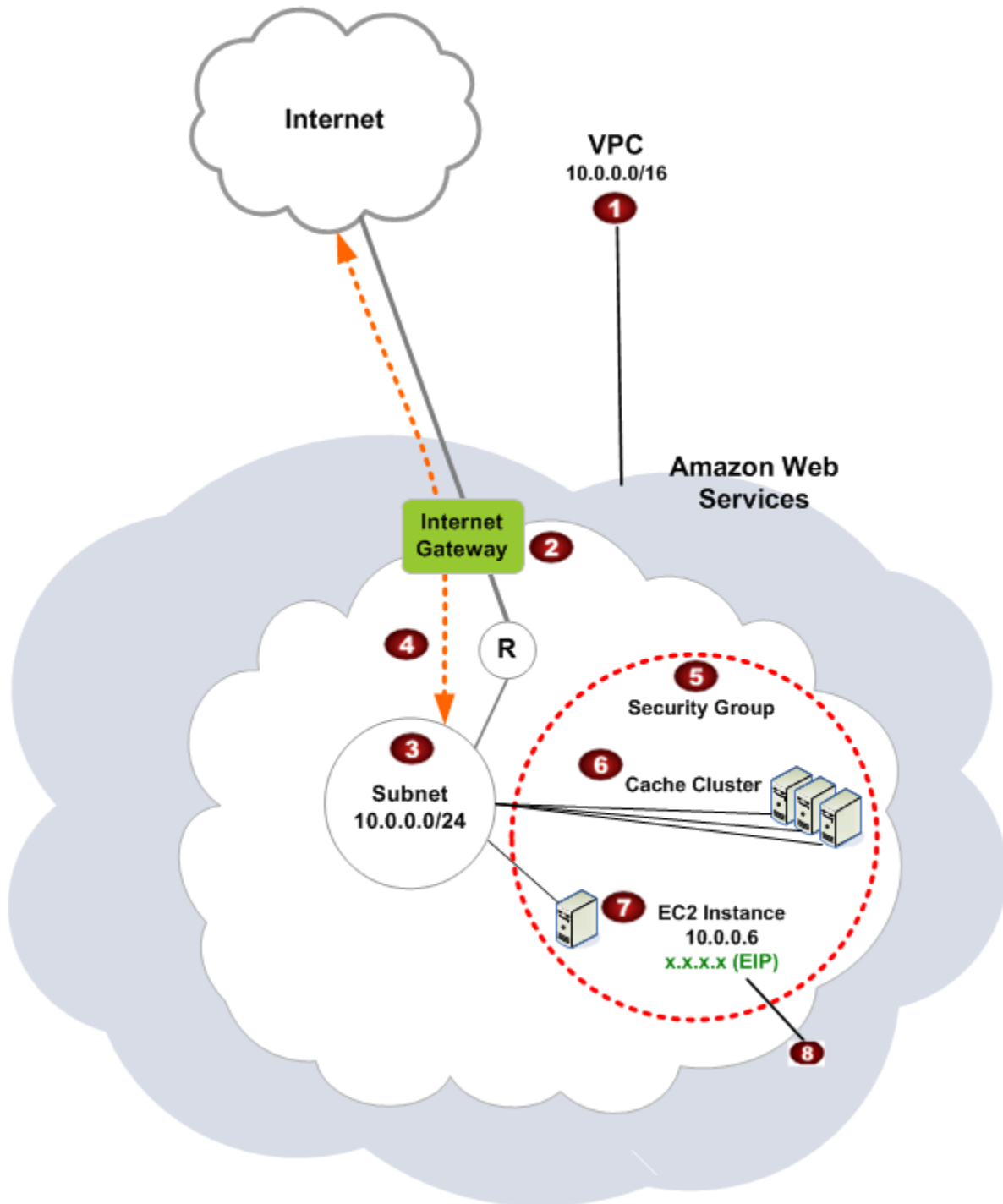
Com a Amazon Virtual Private Cloud, você pode criar uma rede virtual na AWS nuvem que se assemelha muito a um data center tradicional. Você pode configurar sua AmazonVPC, incluindo selecionar seu intervalo de endereços IP, criar sub-redes e configurar tabelas de rotas, gateways de rede e configurações de segurança.

A funcionalidade básica do ElastiCache é a mesma em uma nuvem privada virtual; ElastiCache gerencia atualizações de software, patches, detecção de falhas e recuperação, independentemente de seus clusters estarem implantados dentro ou fora da Amazon. VPC

ElastiCache os nós de cache implantados fora de uma Amazon VPC recebem um endereço IP para o qual o DNS endpoint/nome é resolvido. Isso fornece conectividade a partir de instâncias do Amazon Elastic Compute Cloud (AmazonEC2). Quando você executa um ElastiCache cluster em uma sub-rede VPC privada da Amazon, cada nó de cache recebe um endereço IP privado dentro dessa sub-rede.

### Visão geral de ElastiCache em uma Amazon VPC

O diagrama e a tabela a seguir descrevem o VPC ambiente da Amazon, junto com ElastiCache clusters e EC2 instâncias da Amazon que são lançados na AmazonVPC.



1

A Amazon VPC é uma parte isolada da AWS nuvem à qual é atribuído seu próprio bloco de endereços IP.

2

Um gateway de Internet conecta sua Amazon VPC diretamente à Internet e fornece acesso a outros AWS recursos, como o Amazon Simple Storage Service (Amazon S3), que estão sendo executados fora da sua Amazon. VPC

**3**

Uma VPC sub-rede da Amazon é um segmento do intervalo de endereços IP de uma Amazon em VPC que você pode isolar AWS recursos de acordo com suas necessidades operacionais e de segurança.

**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 VPC de segurança da Amazon controla o tráfego de entrada e saída para seus ElastiCache clusters e instâncias da AmazonEC2.

**6**

Você pode iniciar um ElastiCache cluster na sub-rede. Os nós de cache possuem endereços IP privados a partir do intervalo de endereços da sub-rede.

**7**

Você também pode iniciar EC2 instâncias da Amazon na sub-rede. Cada EC2 instância da Amazon tem um endereço IP privado do intervalo de endereços da sub-rede. A EC2 instância da Amazon pode se conectar a qualquer nó de cache na mesma sub-rede.

**8**

Para que uma EC2 instância da Amazon em sua Amazon VPC possa ser acessada pela Internet, você precisa atribuir um endereço público estático chamado endereço IP elástico à instância.

## Pré-requisitos

Para criar um ElastiCache cluster dentro de uma AmazonVPC, sua Amazon VPC deve atender aos seguintes requisitos:

- A Amazon VPC deve permitir EC2 instâncias Amazon não dedicadas. Você não pode usar ElastiCache em uma Amazon VPC configurada para locação de instâncias dedicadas.

- Um grupo de sub-rede de cache deve ser definido para sua AmazonVPC. ElastiCache usa esse grupo de sub-redes de cache para selecionar uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus VPC endpoints ou nós de cache.
- CIDRs blocos de cada sub-rede devem ser grandes o suficiente para fornecer endereços IP extras ElastiCache para uso durante as atividades de manutenção.

## Roteamento e segurança

Você pode configurar o roteamento em sua Amazon VPC para controlar onde o tráfego flui (por exemplo, para o gateway da Internet ou gateway privado virtual). Com um gateway de Internet, sua Amazon VPC tem acesso direto a outros AWS recursos que não estão sendo executados na sua AmazonVPC. Se você optar por ter apenas um gateway privado virtual com uma conexão com a rede local da sua organização, poderá rotear o tráfego vinculado à Internet pelo VPN e usar políticas de segurança e firewall locais para controlar a saída. Nesse caso, você incorre em cobranças adicionais de largura de banda ao acessar AWS recursos pela Internet.

Você pode usar os grupos VPC de segurança da Amazon para ajudar a proteger os ElastiCache clusters e as EC2 instâncias da Amazon na sua AmazonVPC. Os security groups atuam como um firewall no nível da instância e não no nível da sub-rede.

### Note

É altamente recomendável que você use DNS nomes para se conectar aos nós de cache, pois o endereço IP subjacente pode mudar.

## VPC Documentação da Amazon

VPCA Amazon tem seu próprio conjunto de documentação para descrever como criar e usar sua AmazonVPC. A tabela a seguir fornece links para os VPC guias da Amazon.

Descrição	Documentação
Como começar a usar a Amazon VPC	<a href="#">Começando com a Amazon VPC</a>
Como usar a Amazon VPC por meio do AWS Management Console	<a href="#">Guia VPC do usuário da Amazon</a>

Descrição	Documentação
Descrições completas de todos os VPC comandos da Amazon	<a href="#">Referência da linha de EC2 comando</a> da Amazon (os VPC comandos da Amazon são encontrados na EC2 referência da Amazon)
Descrições completas das VPC API operações, tipos de dados e erros da Amazon	<a href="#">EC2APIReferência da Amazon</a> (as VPC API operações da Amazon são encontradas na EC2 referência da Amazon)
Informações para o administrador da rede que precisa configurar o gateway na sua extremidade de de uma IPsec VPN conexão opcional	<a href="#">O que AWS Site-to-Site VPN é</a>

Para obter informações mais detalhadas sobre a Amazon Virtual Private Cloud, consulte [Amazon Virtual Private Cloud](#).

## Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC

A Amazon ElastiCache oferece suporte aos seguintes cenários para acessar um cache em uma AmazonVPC:

### Sumário

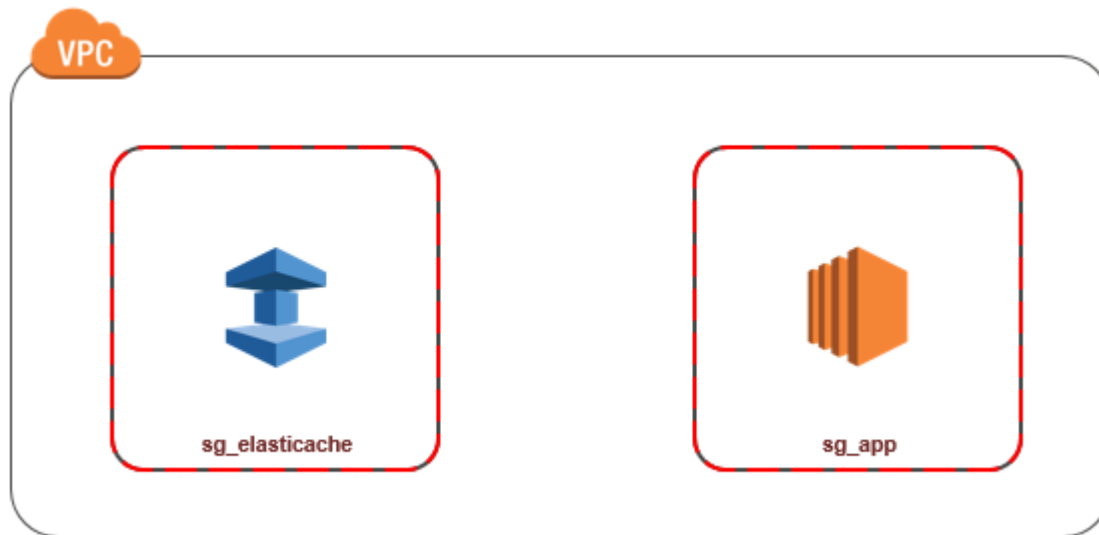
- [Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão na mesma Amazon VPC](#)
- [Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em uma Amazon diferente VPCs](#)
  - [Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em uma Amazon diferente VPCs na mesma região](#)
    - [Uso do Transit Gateway](#)
  - [Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em diferentes Amazon VPCs em diferentes regiões](#)
    - [Usando o Transit VPC](#)
- [Acessando um ElastiCache cache a partir de um aplicativo em execução no data center do cliente](#)
  - [Acessando um ElastiCache cache a partir de um aplicativo executado no data center do cliente usando VPN conectividade](#)
  - [Acessando um ElastiCache cache a partir de um aplicativo executado no data center do cliente usando o Direct Connect](#)

Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão na mesma Amazon VPC

O caso de uso mais comum é quando um aplicativo implantado em uma EC2 instância precisa se conectar a um cache na mesmaVPC.

O diagrama a seguir ilustra esse cenário.





A maneira mais simples de gerenciar o acesso entre EC2 instâncias e caches da mesma forma VPC é fazer o seguinte:

1. Crie um grupo VPC de segurança para seu cache. Esse grupo de segurança pode ser usado para restringir o acesso ao cache. Por exemplo, você pode criar uma regra personalizada para esse grupo de segurança que permita o TCP acesso usando a porta que você atribuiu ao cache quando o criou e um endereço IP que você usará para acessar o cache.

A porta padrão dos caches Memcached é 11211.

A porta padrão para caches Valkey e OSS Redis é 6379

2. Crie um grupo VPC de segurança 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 VPC da tabela de roteamento. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o TCP acesso à EC2 instância pela porta 22.
3. Crie regras personalizadas no grupo de segurança do seu cache 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 caches.

#### Note

Se você estiver planejando usar [Zonas locais](#), verifique se você as habilitou. Quando você cria um grupo de sub-redes nessa zona local, ele VPC é estendido para essa zona local e você VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona

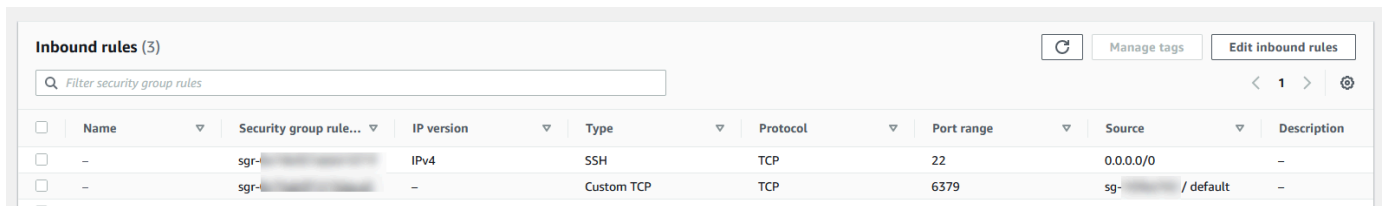
de disponibilidade. Todos os gateways e tabelas de rotas relevantes serão ajustados automaticamente.

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

1. Faça login no AWS Management Console e abra o VPC console da Amazon 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 TCPRegra 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 para OSS caches e grupos de replicação Valkey e Redis 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.



<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-
<input type="checkbox"/>	-	sg-...	-	-	-	-	-	-

Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em uma Amazon diferente VPCs

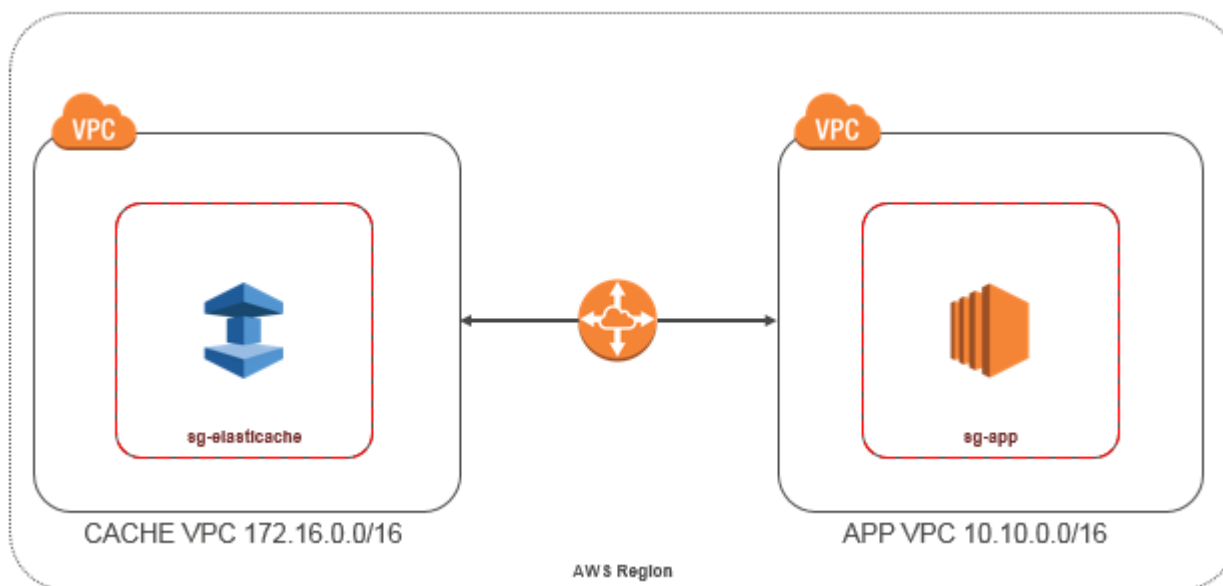
Quando seu cache está em uma EC2 instância VPC diferente da que você está usando para acessá-lo, há várias maneiras de acessar o cache. Se o cache e a EC2 instância estiverem em regiões diferentes VPCs, mas na mesma região, você poderá usar o VPC peering. Se o cache e a EC2 instância estiverem em regiões diferentes, você poderá criar VPN conectividade entre regiões.

## Tópicos

- [Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em uma Amazon diferente VPCs na mesma região](#)
- [Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em diferentes Amazon VPCs em diferentes regiões](#)

Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em uma Amazon diferente VPCs na mesma região

O diagrama a seguir ilustra o acesso a um cache por uma EC2 instância da Amazon em uma Amazon diferente VPC na mesma região usando uma conexão de VPC peering da Amazon.



Cache acessado por uma EC2 instância da Amazon em uma Amazon VPC diferente na mesma região - Conexão de VPC emparelhamento

Uma conexão VPC de peering é uma conexão de rede entre duas VPCs que permite rotear o tráfego entre elas usando endereços IP privados. As instâncias em qualquer uma das VPC podem se comunicar umas com as outras como se estivessem na mesma rede. Você pode criar uma conexão VPC de peering entre sua própria Amazon VPCs ou com uma Amazon VPC em outra AWS conta dentro de uma única região. Para saber mais sobre o VPC peering da Amazon, consulte a [VPC documentação](#).

**Note**

DNSa resolução de nomes pode falhar no peeringVPCs, dependendo das configurações aplicadas ao. ElastiCache VPC Para resolver isso, ambos VPCs devem estar habilitados para DNS nomes de host e DNS resolução. Para obter mais informações, consulte [Habilitar DNS resolução para uma conexão de VPC emparelhamento](#).

Para acessar um cache em uma Amazon diferente VPC por meio de peering

1. Certifique-se de que os dois VPCs não tenham um intervalo de IP sobreposto ou você não conseguirá emparelhá-los.
2. Veja os doisVPCs. Para obter mais informações, consulte [Criação e aceitação de uma conexão de VPC peering da Amazon](#).
3. Atualize sua tabela de roteamento. Para obter mais informações, consulte [Atualizando suas tabelas de rotas para uma conexão de VPC emparelhamento](#)

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

VPCtabela de roteamento

4. Modifique o grupo de segurança do seu ElastiCache cache para permitir a conexão de entrada do grupo de segurança do aplicativo no VPC peering. Para obter mais informações, consulte [Grupos de VPC segurança de pares de referência](#).

O acesso a um cache por meio de uma conexão de emparelhamento implicará custos adicionais de transferência de dados.

## Uso do Transit Gateway

Um gateway de trânsito permite que você conecte VPCs e VPN conecte na mesma AWS região e roteie o tráfego entre elas. Um gateway de trânsito funciona em várias AWS contas, e você pode usar o AWS Resource Access Manager para compartilhar seu gateway de trânsito com outras contas. Depois de compartilhar um gateway de trânsito com outra AWS conta, o proprietário da conta pode anexá-lo VPCs ao seu gateway de trânsito. Um usuário de qualquer uma das contas pode excluir o anexo a qualquer momento.

Você pode habilitar o multicast em um gateway de trânsito e, em seguida, criar um domínio multicast do gateway de trânsito que permita que o tráfego multicast seja enviado da sua origem multicast para membros do grupo multicast por meio de VPC anexos que você associa ao domínio.

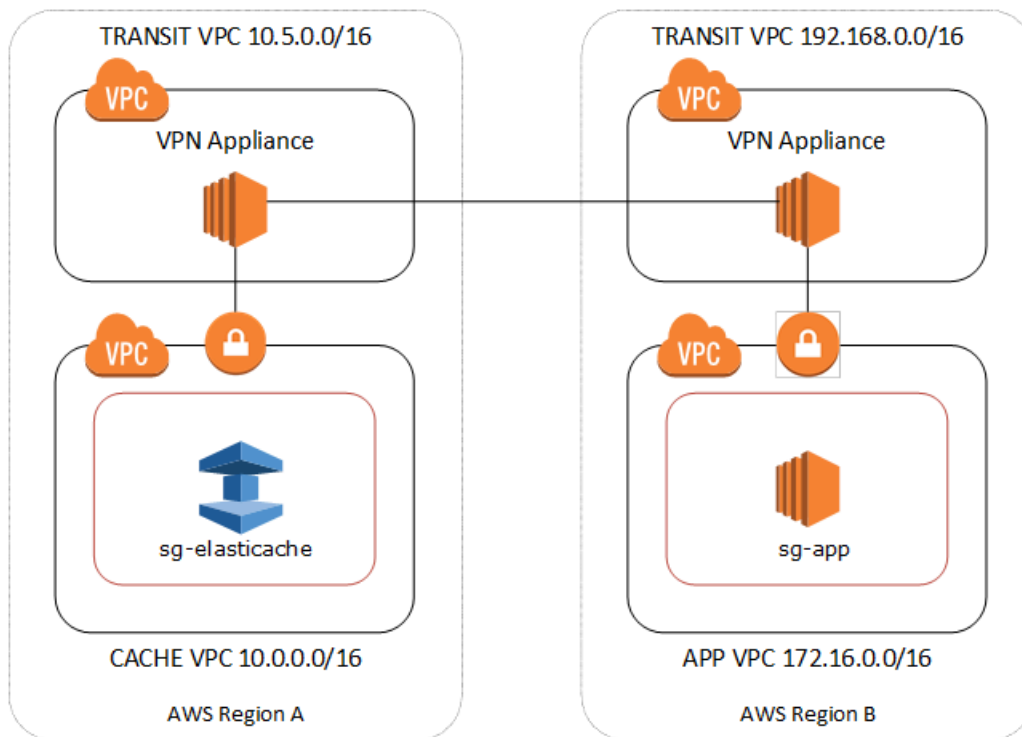
Você também pode criar um anexo de conexão de emparelhamento entre gateways de trânsito em diferentes AWS regiões. Isso permite que você roteie o tráfego entre os anexos dos gateways de trânsito em regiões diferentes.

Para obter mais informações, consulte [Gateways de trânsito](#).

Acessando um ElastiCache cache quando ele e a EC2 instância da Amazon estão em diferentes Amazon VPCs em diferentes regiões

### Usando o Transit VPC

Uma alternativa ao uso do VPC peering, outra estratégia comum para conectar várias redes geograficamente dispersas VPCs e remotas, é criar um trânsito VPC que sirva como um centro de trânsito de rede global. Um trânsito VPC simplifica o gerenciamento da rede e minimiza o número de conexões necessárias para conectar redes múltiplas VPCs e 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.



## Conectando-se entre diferentes VPCs regiões

Depois que o Transit Amazon VPC é estabelecido, um aplicativo implantado em um “raio” VPC em uma região pode se conectar a um ElastiCache cache em um “VPCraio” em outra região.

Para acessar um cache em uma região diferente VPC dentro de uma AWS região diferente

1. Implemente uma VPC solução de transporte público. Para obter mais informações, consulte [Transit Gateway da AWS](#).
2. Atualize as tabelas VPC de roteamento no aplicativo e no cache VPCs para rotear o tráfego através do VGW (Virtual Private Gateway) e do VPN dispositivo. No caso de roteamento dinâmico com Border Gateway Protocol (BGP), suas rotas podem ser propagadas automaticamente.
3. Modifique o grupo de segurança do seu ElastiCache cache para permitir a conexão de entrada a partir do intervalo de IP das instâncias do aplicativo. 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.

## Acessando um ElastiCache cache a partir de um aplicativo em execução no data center do cliente

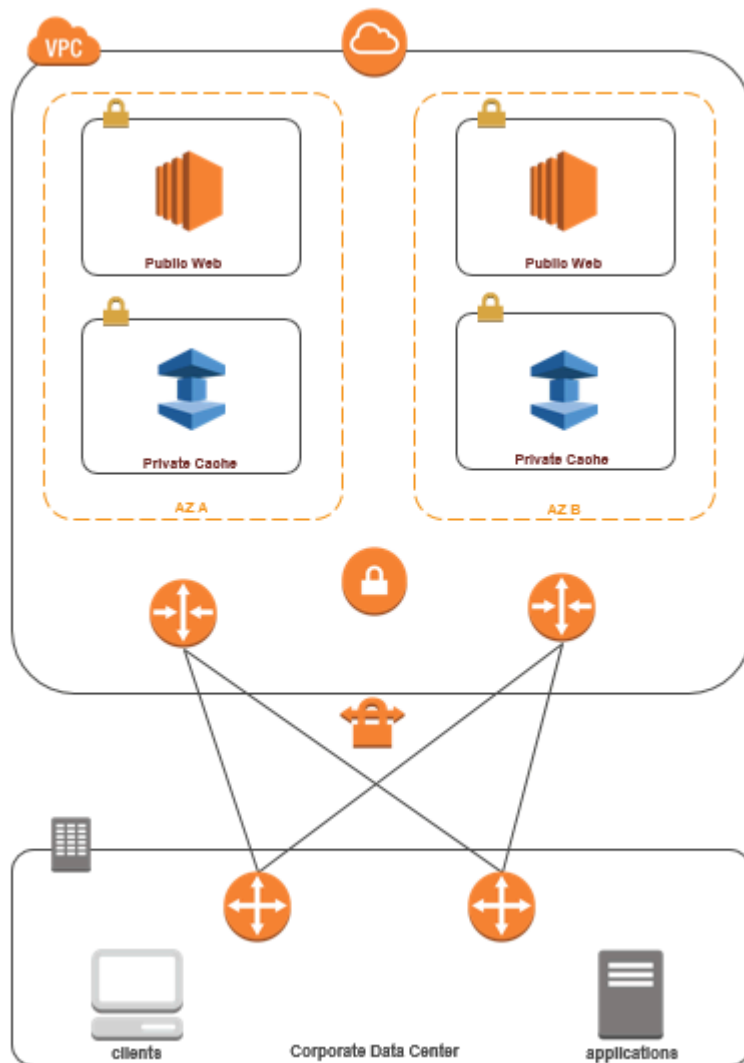
Outro cenário possível é uma arquitetura híbrida em que clientes ou aplicativos no data center do cliente podem precisar acessar um ElastiCache cache noVPC. Esse cenário também é suportado, desde que haja conectividade entre os clientes VPC e o data center por meio do Direct Connect VPN ou do Direct Connect.

### Tópicos

- [Acessando um ElastiCache cache a partir de um aplicativo executado no data center do cliente usando VPN conectividade](#)
- [Acessando um ElastiCache cache a partir de um aplicativo executado no data center do cliente usando o Direct Connect](#)

## Acessando um ElastiCache cache a partir de um aplicativo executado no data center do cliente usando VPN conectividade

O diagrama a seguir ilustra o acesso a um ElastiCache cache a partir de um aplicativo em execução na rede corporativa usando VPN conexões.



Conectando-se a ElastiCache partir do seu data center por meio de um VPN

Para acessar um cache em um aplicativo local por meio VPC da conexão VPN

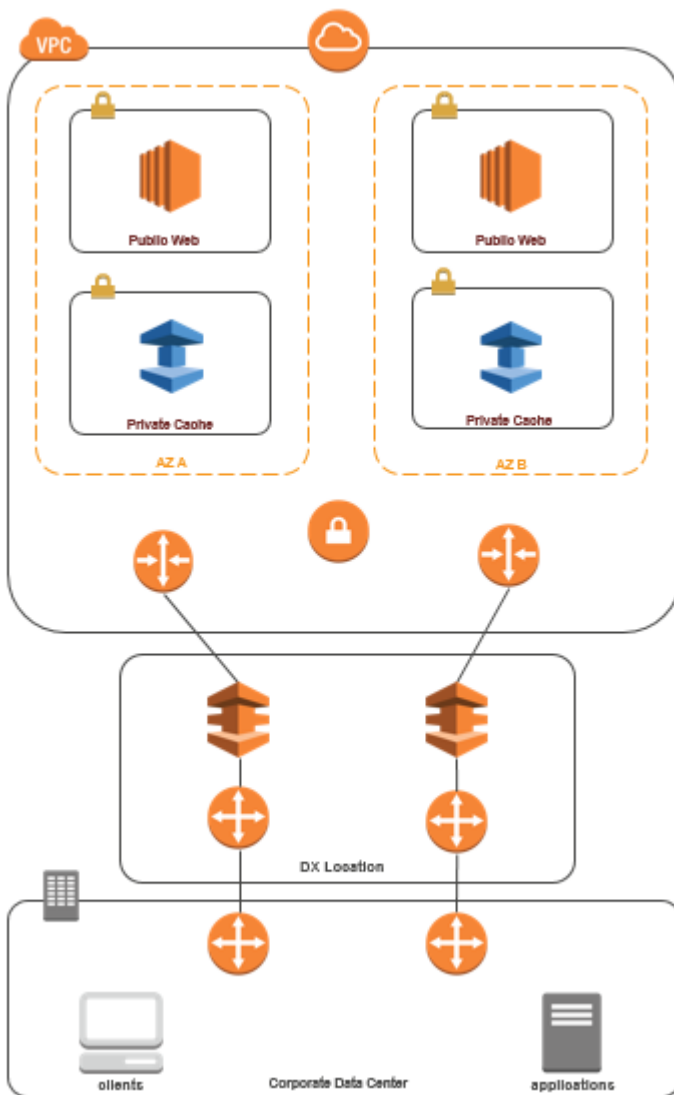
1. Estabeleça VPN conectividade adicionando um gateway privado virtual de hardware ao seu VPC. Para obter mais informações, consulte [Adicionando um gateway privado virtual de hardware ao seu VPC](#).
2. Atualize a tabela VPC de roteamento da sub-rede em que seu ElastiCache cache está implantado para permitir o tráfego do seu servidor de aplicativos local. No caso de roteamento dinâmico, BGP suas rotas podem ser propagadas automaticamente.
3. Modifique o Grupo de Segurança do seu ElastiCache cache para permitir a conexão de entrada dos servidores de aplicativos locais.



O acesso a um cache por meio de uma VPN conexão introduzirá latências de rede e custos adicionais de transferência de dados.

Acessando um ElastiCache cache a partir de um aplicativo executado no data center do cliente usando o Direct Connect

O diagrama a seguir ilustra o acesso a um ElastiCache cache a partir de um aplicativo em execução na rede corporativa usando o Direct Connect.



Conectando-se a ElastiCache partir do seu data center via Direct Connect

Para acessar um ElastiCache cache de um aplicativo em execução na sua rede usando o Direct Connect

1. Estabeleça a conectividade Direct Connect. Para obter mais informações, consulte [Introdução ao AWS Direct Connect](#).
2. Modifique o Grupo de Segurança do seu ElastiCache cache para permitir a conexão de entrada dos servidores de aplicativos locais.

O acesso a um cache por meio de uma conexão DX pode introduzir latências de rede e taxas adicionais de transferência de dados.

## Criando uma nuvem privada virtual (VPC)

Neste exemplo, você cria uma Amazon VPC com uma sub-rede privada para cada zona de disponibilidade.

### Criando um Amazon VPC (Console)

1. Faça login no AWS Management Console e abra o VPC console da Amazon em <https://console.aws.amazon.com/vpc/>.
2. No VPC painel, escolha Criar VPC.
3. Em Recursos para criar, escolha VPCe muito 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 privadas 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 AmazonVPC.

7. Crie um grupo de VPC segurança da Amazon. Você usará esse grupo para seu cluster de cache e sua EC2 instância da Amazon.
  - a. No painel de navegação do console de VPC gerenciamento da Amazon, escolha Security Groups.
  - b. Escolha Criar grupo de segurança.
  - c. Digite um nome e uma descrição do seu security group nas caixas correspondentes. Na VPCcaixa, escolha o identificador da sua AmazonVPC.

**Create security group** [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)  
  
Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

**Inbound rules** [Info](#)

This security group has no inbound rules.

**Outbound rules** [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
All traffic	All	All	Custom <input type="text" value="0.0.0.0/0"/>	<input type="text"/>

- 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 Criar uma nova regra, escolha e SSH, em seguida, escolha Adicionar regra.
  - d. Defina os seguintes valores para sua nova regra de entrada para permitir o HTTP acesso:
    - Tipo: HTTP
    - Origem: 0.0.0.0/0

Escolha Apply Rule Changes.

Agora você está pronto para criar um grupo de sub-rede de cache e iniciar um cluster de cache na sua AmazonVPC.

- [Criação de um grupo de sub-redes](#)
- [Criação de um cluster do Memcached \(console\).](#)
- [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\).](#)

## Conectando-se a um cache em execução em uma Amazon VPC

Este exemplo mostra como iniciar uma EC2 instância da Amazon na sua AmazonVPC. Em seguida, você pode fazer login nessa instância e acessar o ElastiCache cache que está sendo executado na AmazonVPC.

### Conectando-se a um cache em execução em um Amazon VPC (Console)

Neste exemplo, você cria uma EC2 instância da Amazon na sua AmazonVPC. Você pode usar essa EC2 instância da Amazon para se conectar aos nós de cache em execução na AmazonVPC.

#### Note

Para obter informações sobre o uso da AmazonEC2, consulte o [Amazon EC2 Getting Started Guide](#) na [EC2documentação da Amazon](#).

Para criar uma EC2 instância da Amazon na sua Amazon VPC usando o EC2 console da Amazon

1. Faça login no AWS Management Console 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 Choose an Amazon Machine Image (AMI), escolha o Amazon Linux de 64 bits eAMI, em seguida, escolha 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 Rede, escolha sua AmazonVPC.
  - 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

Purchasing option  Request Spot Instances

Network

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 example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
<input type="checkbox"/> sg-1a3d2178	default	default VPC security group
<input checked="" type="checkbox"/> sg-f13d2193	my-vpc-security-group	Testing

Escolha o nome do seu grupo de VPC segurança da Amazon e, em seguida, escolha Revisar 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 VPC console da Amazon 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 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 de cache (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 um VPC com o Memcached ou o Redis.

Conectando-se a um VPC com o Memcached

1. Acesse o ElastiCache console em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint de um dos nós em seu cluster de cache. 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
```

## Conectando-se a um VPC com o Redis

1. Acesse o ElastiCache console em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint de um dos nós em seu cluster de cache. Para obter mais informações, consulte [Como encontrar pontos de extremidade de conexão](#) para o Redis.

- 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
```

- Acesse o ElastiCache console em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint de um dos nós em seu cluster de cache. Para obter mais informações, consulte [Encontrar pontos de extremidade de conexão](#) para o RedisOSS.
- 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 APIe VPC endpoints de interface ( )AWS PrivateLink

Você pode estabelecer uma conexão privada entre seus ElastiCache API endpoints VPC e a Amazon criando um VPCendpoint de interface. Os endpoints de interface são alimentados por [AWS PrivateLink](#). AWS PrivateLink permite que você acesse de forma privada ElastiCache API as operações da Amazon sem um gateway de internet, NAT dispositivo, VPN conexão ou conexão AWS Direct Connect.

As suas instâncias VPC não precisam de endereços IP públicos para se comunicar com os ElastiCache API endpoints da Amazon. Suas instâncias também não precisam de endereços IP

públicos para usar nenhuma das ElastiCache API operações disponíveis. O tráfego entre você 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 \(AWS PrivateLink\)](#) no Amazon VPC User Guide.
- Para obter mais informações sobre ElastiCache API operações, consulte [ElastiCache APIoperações](#).

Depois de criar um VPC endpoint de interface, se você habilitar DNS nomes de host [privados](#) para o endpoint, o endpoint padrão ElastiCache (<https://elasticache.Region.amazonaws.com>) resolve para seu endpoint. VPC Se você não habilitar DNS nomes de host privados, a Amazon VPC fornecerá um nome de DNS endpoint que você pode usar no seguinte formato:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Para obter mais informações, consulte [Interface VPC Endpoints \(AWS PrivateLink\)](#) no Guia do VPC usuário da Amazon. ElastiCache suporta fazer chamadas para todas as suas [APIações](#) dentro do seuVPC.

#### Note

DNSNomes de host privados só podem ser habilitados para um VPC endpoint no. VPC Se você quiser criar um VPC endpoint adicional, o DNS nome do host privado deve ser desabilitado para ele.

## Considerações sobre endpoints VPC

Antes de configurar um VPC endpoint de interface para ElastiCache API endpoints da Amazon, certifique-se de revisar as [propriedades e limitações do endpoint da interface](#) no Guia do usuário da Amazon VPC. Todas as ElastiCache API operações relevantes para gerenciar ElastiCache os recursos da Amazon estão disponíveis a partir do seu VPC uso AWS PrivateLink.

VPCpolíticas de endpoint são suportadas para ElastiCache API endpoints. Por padrão, o acesso total às ElastiCache API operações é permitido por meio do endpoint. Para obter mais informações, consulte [Controle de acesso a serviços com VPC endpoints](#) no Guia do VPC usuário da Amazon.

## Criando um VPC endpoint de interface para o ElastiCache API

Você pode criar um VPC endpoint para a Amazon ElastiCache API usando o VPC console da Amazon ou o AWS CLI. Para obter mais informações, consulte [Criação de um endpoint de interface](#) no Guia do VPC usuário da Amazon.

Depois de criar um VPC endpoint de interface, você pode habilitar DNS nomes de host privados para o endpoint. Quando você fizer isso, o ElastiCache endpoint padrão da Amazon (<https://elasticache.Region.amazonaws.com>) resolve para seu endpoint. VPC Para as AWS regiões da China (Pequim) e da China (Ningxia), você pode fazer API solicitações com o VPC endpoint usando [elasticache.cn-north-1.amazonaws.com.cn](https://elasticache.cn-north-1.amazonaws.com.cn) para Pequim e [elasticache.cn-northwest-1.amazonaws.com.cn](https://elasticache.cn-northwest-1.amazonaws.com.cn) para Ningxia. Para obter mais informações, consulte [Acessando um serviço por meio de um endpoint de interface](#) no Guia do VPC usuário da Amazon.

## Criação de uma política VPC de endpoint para a Amazon ElastiCache API

Você pode anexar uma política de endpoint ao seu VPC endpoint que controla o acesso ao ElastiCache API. A política especifica o seguinte:

- A entidade principal que pode executar ações.
- As ações que podem ser executadas.
- Os recursos sobre os quais as ações podem ser realizadas.

Para obter mais informações, consulte [Controle de acesso a serviços com VPC endpoints](#) no Guia do VPC usuário da Amazon.

Exemplo VPC política de endpoint para ElastiCache API ações com Valkey ou Redis OSS

Veja a seguir um exemplo de uma política de endpoint para o ElastiCache API. Quando anexada a um endpoint, essa política concede acesso às ElastiCache API ações listadas para todos os diretores em todos os recursos.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster",
```

```

    "elasticache:CreateSnapshot"
  ],
  "Resource": "*"
}]
}

```

### Example VPC política de endpoint para ações ElastiCache (Memcached) API

Veja a seguir um exemplo de uma política de endpoint para o ElastiCache API. Quando anexada a um endpoint, essa política concede acesso às ElastiCache API ações listadas para todos os diretores em todos os recursos.

```

{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
  }]
}

```

### Example VPC política de endpoint que nega todo o acesso de uma conta especificada AWS

A seguinte política de VPC endpoint nega AWS a conta **123456789012** todo o acesso aos recursos usando 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 uma coleção de sub-redes (normalmente privadas) que você pode designar para seus clusters autoprojeto em execução em um ambiente Amazon Virtual Private Cloud (VPC).

Se você criar um cluster autoprojeto em uma AmazonVPC, 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 IPv6 CIDR bloco. 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 VPC ambiente Amazon, 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 para seus caches em um VPC. Ao iniciar um cache em um VPC, você precisa 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 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 rede, um grupo de sub-rede padrão estará disponível ou você poderá optar por criar um novo. 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, você será direcionado para criar pilha dupla ou sub-redes. IPv6 Para obter mais informações sobre os tipos de rede, consulte [Network type](#). Para obter mais informações, consulte [Criar uma sub-rede no seu VPC](#).

Os procedimentos a seguir mostram como criar um grupo de sub-redes chamado mysubnetgroup (console) AWS CLI, o e o. ElastiCache API

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 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 VPCID, escolha sua AmazonVPC.
  - d. Todas as sub-redes são escolhidas por padrão. No painel Sub-redes selecionadas, clique em Gerenciar e selecione as Zonas de Disponibilidade ou [Zonas Locais](#) e IDs de suas sub-redes privadas e escolha Escolher.
5. Na mensagem de confirmação exibida, escolha 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.

### 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 uma AmazonVPC. 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 VPC grupo de sub-redes.
- Cluster Valkey ou Redis autônomo — Para iniciar um OSS cluster Valkey ou Redis de nó único, consulte. OSS [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#) Na etapa 7.a (OSSConfigurações avançadas do Redis), escolha um grupo de VPC sub-redes.
- Grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) — Para iniciar um grupo de replicação Valkey ou Redis OSS (modo de cluster desativado) em um, consulte. VPC [Criando um grupo de replicação Valkey ou Redis OSS \(modo de cluster desativado\) do zero](#) Na etapa 7.b (OSSConfigurações avançadas do Redis), escolha um grupo de VPC sub-redes.
- Grupo de replicação Valkey ou Redis OSS (modo de cluster ativado) —. [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#) Na etapa 6.i (OSSConfigurações avançadas do Redis), escolha um grupo de VPC sub-redes.

## Modificação de um grupo de sub-redes

Você pode modificar a descrição de um grupo de sub-redes ou modificar a lista de sub-redes IDs associadas 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 AWS Management Console e abra o ElastiCache console 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 AWS Management Console e abra o ElastiCache console 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. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar ElastiCache os recursos. IAMé um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

## Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)
- [Como a Amazon ElastiCache trabalha com IAM](#)
- [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)
- [Solução de problemas de ElastiCache identidade e acesso da Amazon](#)
- [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, dependendo do trabalho que você faz ElastiCache.

**Usuário do serviço** — Se você usar o ElastiCache serviço para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais ElastiCache recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso no ElastiCache, consulte [Solução de problemas de ElastiCache identidade e acesso da Amazon](#).

**Administrador de serviços** — Se você é responsável pelos ElastiCache recursos da sua empresa, provavelmente tem acesso total ElastiCache a. É seu trabalho determinar quais ElastiCache recursos e recursos seus usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta

página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM com ElastiCache, consulte [Como a Amazon ElastiCache trabalha com IAM](#).

**IAM administrador** — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso ElastiCache. Para ver exemplos de políticas ElastiCache baseadas em identidade que você pode usar em IAM, consulte [Exemplos de políticas baseadas em identidade para a Amazon ElastiCache](#)

## Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login AWS, consulte [Como fazer login Conta da AWS no](#) Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [Assinar AWS API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia AWS IAM Identity Center do usuário e [Uso da autenticação multifator \(MFA\) AWS no](#) Guia do IAM usuário.



## Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

## Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, 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 de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

## Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de

uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um IAM usuário \(em vez de uma função\)](#) no Guia do IAM usuário.

## IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Métodos para assumir uma função](#) no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter informações sobre funções para federação, consulte [Criação de uma função para um provedor de identidade terceirizado](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .
- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte [Acesso a recursos entre contas IAM no Guia](#) do IAM usuário.

- **Acesso entre serviços** — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
- **Sessões de acesso direto (FAS)** — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- **Função de serviço** — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.
- **Função vinculada ao serviço** — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode presumir a função 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 IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.
- **Aplicativos em execução na Amazon EC2** — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte [Quando criar uma IAM função \(em vez de um usuário\)](#) no Guia do IAM usuário.

## Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

### Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. 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 [Criação de IAM políticas no Guia](#) do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolha entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

## Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

## Listas de controle de acesso (ACLs)

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 JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para IAM entidades](#) no Guia IAM do usuário.

- Políticas de controle de serviço (SCPs) — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia IAM do usuário.

## 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 política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

## Como a Amazon ElastiCache trabalha com IAM

Antes de usar IAM para gerenciar o acesso ao ElastiCache, saiba quais IAM recursos estão disponíveis para uso ElastiCache.

IAMrecursos que você pode usar com a Amazon ElastiCache

IAMrecurso	ElastiCache apoio
<a href="#">Políticas baseadas em identidade</a>	Sim
<a href="#">Políticas baseadas em recursos</a>	Não
<a href="#">Ações das políticas</a>	Sim

IAMrecurso	ElastiCache apoio
<a href="#">Atributos de políticas</a>	Sim
<a href="#">Chaves de condição de políticas</a>	Sim
<a href="#">ACLs</a>	Sim
<a href="#">ABAC(tags nas políticas)</a>	Sim
<a href="#">Credenciais temporárias</a>	Sim
<a href="#">Permissões de entidade principal</a>	Sim
<a href="#">Perfis de serviço</a>	Sim
<a href="#">Funções vinculadas a serviço</a>	Sim

Para obter uma visão geral de como ElastiCache e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte [AWS os serviços que funcionam com IAM](#) no Guia do IAM usuário.

## Políticas baseadas em identidade para ElastiCache

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. 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 [Criação de IAM políticas no Guia](#) do IAM usuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

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

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, [consulte Acesso a recursos entre contas IAM no](#) Guia do IAM usuário.

## Ações políticas para ElastiCache

Compatível com ações de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.



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 AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento Resource JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento Resource ou NotResource. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, 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 cada recurso, consulte [Ações definidas pela Amazon ElastiCache](#). ARN

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 AWS JSON políticas 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` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. 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.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte [elementos de IAM política: variáveis e tags](#) no Guia IAM do usuário.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia IAM do usuário.

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

SuportesACLs: 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 JSON de política.

## Controle de acesso baseado em atributos ( ) ABAC com ElastiCache

Suportes ABAC (tags nas políticas): Sim

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

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 oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre ABAC, consulte [O que é ABAC?](#) no Guia do IAM usuário. Para ver um tutorial com etapas de configuração ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAM usuário.

## Usando credenciais temporárias com ElastiCache

Compatível com credenciais temporárias: Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS nesse trabalho IAM](#) no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte [Alternando para uma função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias em IAM](#).

## Permissões de entidade principal entre serviços para o ElastiCache

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).

## Funções de serviço para ElastiCache

Compatível com perfis de serviço: Sim

Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente em IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.

**⚠ Warning**

Alterar as permissões de uma função de serviço pode interromper ElastiCache a funcionalidade. Edite as funções de serviço somente quando ElastiCache fornecer orientação para fazer isso.

## Funções vinculadas a serviços para ElastiCache

Suporte a 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 presumir a função 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 IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte [AWS serviços que funcionam](#) com. IAM Encontre um serviço na tabela que inclua um Yes na coluna Função vinculada 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 funções não têm permissão para criar ou modificar ElastiCache recursos. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte [Criação de IAM políticas no Guia](#) do IAM usuário.

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

- [Melhores práticas de política](#)

- [Usando o ElastiCache console](#)
- [Permitir que usuários visualizem suas próprias permissões](#)

## Melhores práticas 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 obter mais informações, consulte [políticas AWS gerenciadas](#) ou [políticas AWS gerenciadas para funções de trabalho](#) no Guia IAM do usuário.
- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte [Políticas e permissões IAM no](#) Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode 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 AWS CloudFormation. Para obter mais informações, consulte [Elementos IAM JSON da política: Condição](#) no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação da política do IAM Access Analyzer](#) no Guia do IAM Usuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando

API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte [Configurando o API acesso MFA protegido](#) no Guia do IAM usuário.

Para obter mais informações sobre as melhores práticas em IAM, consulte [as melhores práticas de segurança IAM no](#) Guia IAM do usuário.

## 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. Se você criar uma política baseada em identidade que seja mais restritiva do 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 AWS CLI o. ou AWS API o. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Para garantir que usuários e funções ainda possam usar o ElastiCache console, anexe também a política ElastiCache ConsoleAccess ou a política ReadOnly AWS gerenciada às entidades. Para obter mais informações, consulte [Adicionar permissões a um usuário](#) no Guia do IAM usuário.

## Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou. AWS API

```
{
  "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 ElastiCache identidade e acesso da Amazon

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com ElastiCache e IAM

### Tópicos

- [Não estou autorizado a realizar uma ação em ElastiCache](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha AWS conta acessem meus ElastiCache recursos](#)

### Não estou autorizado a realizar uma ação em ElastiCache

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. 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 elasticache: `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 e permitir o acesso ao recurso `my-example-widget` usando a ação elasticache: `GetWidget`.

## Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você transfira uma função para ElastiCache o.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar 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 precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Quero permitir que pessoas fora da minha AWS conta acessem meus ElastiCache recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a 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 é ElastiCache compatível com esses recursos, consulte [Como a Amazon ElastiCache trabalha com IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outro Conta da AWS de sua propriedade](#) no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

## 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 \(IAMpolíticas\) 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 da conta pode anexar políticas de permissões às IAM identidades (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 [as IAM melhores práticas](#) no Guia IAM do usuário.

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

- Usuários e grupos em AWS IAM Identity Center:

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

- Usuários gerenciados IAM por meio de um provedor de identidade:

Crie um perfil para a federação de identidades. Siga as instruções em [Criação de uma função para um provedor de identidade terceirizado \(federação\)](#) no Guia IAM do usuário.

- IAMusuários:

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

### Tópicos

- [ElastiCache Recursos e operações da Amazon](#)
- [Informações sobre propriedade de recursos](#)
- [Gerenciamento de acesso aos recursos](#)

- [AWS políticas gerenciadas para a Amazon ElastiCache](#)
- [Usando políticas baseadas em identidade \(IAMpolíticas\) para a Amazon ElastiCache](#)
- [Permissões em nível de recurso](#)
- [Uso de chaves de condição](#)
- [Usando funções vinculadas a serviços para a Amazon ElastiCache](#)
- [ElastiCache APIpermissões: referência de ações, recursos e condições](#)

## ElastiCache Recursos e operações da Amazon

Para ver uma lista dos tipos de ElastiCache recursos e seusARNs, consulte [Recursos definidos pela Amazon ElastiCache](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Ações definidas pela Amazon ElastiCache](#). ARN

## 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 IAM usuário ou uma IAM função). 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 de cache.
- Suponha que você crie um IAM usuário em sua AWS conta e conceda permissões para criar um cluster de cache para esse usuário. Nesse caso, o usuário pode criar um cluster de cache. No entanto, sua AWS conta, à qual o usuário pertence, é proprietária do recurso de cluster de cache.
- Suponha que você crie uma IAM função na sua AWS conta com permissões para criar um cluster de cache. Nesse caso, qualquer pessoa que possa assumir a função poderá criar um cluster de cache. Sua AWS conta, à qual a função pertence, é proprietária do recurso de cluster de cache.

## Gerenciamento de acesso aos recursos

A política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções disponíveis para a criação das políticas de permissões.

**Note**

Esta seção discute o uso IAM no contexto da Amazon ElastiCache. Ele não fornece informações detalhadas sobre o IAM serviço. Para obter a IAM documentação completa, consulte [O que é IAM?](#) no Guia do IAM usuário. Para obter informações sobre a IAM sintaxe e as descrições das [AWS IAM políticas](#), consulte [Referência de políticas](#) no Guia do IAM usuário.

As políticas anexadas a uma IAM identidade são chamadas de políticas baseadas em identidade (IAM políticas). As políticas anexadas a um recurso são chamadas de políticas baseadas em recursos.

**Tópicos**

- [Políticas baseadas em identidade \(políticas\) IAM](#)
- [Especificar elementos da política: ações, efeitos, recursos e entidades principais](#)
- [Especificar condições em uma política](#)

**Políticas baseadas em identidade (políticas) IAM**

Você pode anexar políticas às IAM identidades. 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 de cache, um grupo de parâmetros ou um grupo de segurança.
- Anexar uma política de permissões a uma função (conceder permissões entre contas) — Você pode anexar uma política de permissões baseada em identidade a uma IAM função 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. O administrador da Conta A cria uma IAM função e anexa uma política de permissões à função que concede permissões sobre recursos na 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 IAM como delegar permissões, consulte [Gerenciamento de acesso](#) no Guia do IAM usuário.

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 API ações. Essa abordagem também é chamada de permissões no nível do recurso.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DescribeCacheClusters",
    "Effect": "Allow",
    "Action": [
      "elasticache:DescribeCacheClusters"],
    "Resource": resource-arn
  ]
}
```

Para obter mais informações sobre o uso de políticas baseadas em identidade com ElastiCache, consulte. [Usando políticas baseadas em identidade \(IAM políticas\) para a Amazon ElastiCache](#) Para obter mais informações sobre usuários, grupos, funções e permissões, consulte [Identidades \(usuários, grupos e funções\)](#) no Guia do IAM usuário.

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 API operações (consulte [Ações](#)). Para conceder permissões para essas API operações, ElastiCache define 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 API operação 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) para identificar o recurso ao qual 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 especificado `Effect`, 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.
- **Principal** — Em políticas baseadas em identidade (IAM políticas), o usuário ao qual a política está vinculada é o principal implícito. 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 das IAM políticas, consulte a [Referência AWS IAM de políticas](#) no Guia do IAM usuário.

Para ver uma tabela mostrando todas as ElastiCache API ações da Amazon, consulte [ElastiCache API permissões: referência de ações, recursos e condições](#).

### Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da IAM política para especificar as condições em que uma política deve entrar em vigor. Por exemplo, é recomendável aplicar uma política somente após uma data específica. Para obter mais informações sobre a especificação de condições em um idioma de política, consulte [Condição](#) no Guia do IAM Usuário.

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 amplas, consulte [Chaves disponíveis para condições](#) no Guia do IAM usuário.

## AWS políticas gerenciadas para a Amazon ElastiCache

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. 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 [políticas gerenciadas pelo cliente da](#) 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. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas API operações são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte [políticas AWS gerenciadas](#) no Guia IAM do usuário.

### AWS política gerenciada: ElastiCacheServiceRolePolicy

Você não pode se vincular ElastiCacheServiceRolePolicy às suas IAM entidades. 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 VPC endpoints (para caches sem servidor), Elastic Network Interfaces (ENIs) (para clusters autoprotetidos) 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 IAM console e [ElastiCacheServiceRolePolicy](#) no Guia de referência de políticas AWS gerenciadas.



## AWS política gerenciada: AmazonElastiCacheFullAccess

Você pode anexar a `AmazonElastiCacheFullAccess` política às suas IAM identidades.

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 VPC endpoints (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 SNS tópicos para a funcionalidade de notificação no console.

Você pode encontrar a [AmazonElastiCacheFullAccess](#) política no IAM console e [AmazonElastiCacheFullAccess](#) no Guia de referência de políticas AWS gerenciadas.

## AWS política gerenciada: AmazonElastiCacheReadOnlyAccess

Você pode anexar a `AmazonElastiCacheReadOnlyAccess` política às suas IAM identidades.

Essa política permite que os diretores tenham acesso somente para leitura ao ElastiCache uso do AWS Management Console:

- `elasticache`— Acesso somente para leitura `DescribeAPIs`.

Você pode encontrar a [AmazonElastiCacheReadOnlyAccess](#) política no IAM console 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 RSS feed na página Histórico do ElastiCache documento.

Alteração	Descrição	Data
<a href="#">AmazonElastiCacheFullAccess</a> : atualizar 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
<a href="#">ElastiCacheServiceRolePolicy</a> : atualizar 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 (IAMpolíticas) para a Amazon ElastiCache

Este tópico fornece exemplos de políticas baseadas em identidade nas quais um administrador de conta pode anexar políticas de permissões a IAM identidades (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](#)

Veja a seguir um exemplo de uma política de permissões ao usar o RedisOSS.

```
{
  "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"
    }
  ]
}
```

Veja a seguir um exemplo de uma política de permissões ao usar o Memcached.

```
{
  "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 IAM ação (`iam:PassRole`) no nome da IAM função especificado no final do Resource valor.

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 uma IAM função, o principal identificado na política de confiança da função obtém as permissões.

Para ver uma tabela mostrando todas as ElastiCache API ações da Amazon e os recursos aos quais elas se aplicam, consulte [ElastiCache API permissões: 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 ligue IAM CreateServiceLinkedRole API](#)). 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 IAM usuários e grupos, consulte [Como criar seu primeiro grupo de IAM usuários e administradores](#) no Guia do IAM usuário.

#### Important

Sempre teste suas IAM políticas minuciosamente antes de usá-las na 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 clusters de ElastiCache cache. 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 ElastiCache API ações](#)
- [Exemplo 4: Permitir que um usuário ligue IAM CreateServiceLinkedRole API](#)
- [Exemplo 5: Permitir que um usuário se conecte ao cache sem servidor usando autenticação 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.

```
{
  "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 os ElastiCache eventos. 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.

```

{
  "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 ElastiCache API ações

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.

```

{
  "Version": "2012-10-17",

```

```

    "Statement": [{
      "Sid": "EAllowAll",
      "Effect": "Allow",
      "Action": [
        "elasticache:*"
      ],
      "Resource": "*"
    }
  ]
}

```

#### Exemplo 4: Permitir que um usuário ligue IAM CreateServiceLinkedRole API

A política a seguir permite que o usuário chame IAM CreateServiceLinkedRole API o. Recomendamos que você conceda esse tipo de política de permissões ao usuário que invoca operações ElastiCache mutativas.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWS ServiceName": "elasticache.amazonaws.com"
        }
      }
    }
  ]
}

```

#### Exemplo 5: Permitir que um usuário se conecte ao cache sem servidor usando autenticação IAM

A política a seguir permite que qualquer usuário se conecte a qualquer cache sem servidor usando IAM autenticação entre 2023-04-01 e 2023-06-30.

```

{

```

```
"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 IAM política. Muitas ElastiCache API ações oferecem suporte a um tipo de recurso que varia de acordo com o comportamento da ação. Cada declaração IAM de política concede permissão para uma ação que é executada 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 API ações, você pode restringir os recursos que um usuário pode modificar especificando o Amazon Resource Name (ARN) de um recurso ou um ARN padrão que corresponda a vários recursos. Para restringir as permissões por recurso, especifique o recurso por ARN.

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 cada recurso, consulte [Ações definidas pela Amazon ElastiCache](#). ARN

## Exemplos

- [Exemplo 1: permitir que um usuário tenha acesso total a tipos ElastiCache de recursos específicos](#)



- [Exemplo 2: negue a um usuário acesso a um cache sem servidor.](#)

Exemplo 1: permitir que um usuário tenha acesso total a tipos ElastiCache de recursos específicos

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 condições que determinam como uma IAM política entra em vigor. Em ElastiCache, você pode usar o `Condition` elemento de uma JSON política para comparar chaves no contexto da solicitação com valores de chave que você especifica em sua política. Para obter mais informações, consulte [elementos IAM JSON da política: 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](#).

## Especificação de condições: uso de chaves de condição

Para implementar um controle refinado, você escreve uma política de IAM permissões que especifica as condições para controlar um conjunto de parâmetros individuais em determinadas solicitações. Em seguida, você aplica a política aos IAM usuários, grupos ou funções que você cria usando o IAM console.

Para aplicar uma condição, você adiciona as informações da condição à declaração IAM de política. No exemplo a seguir, você especifica a condição de que qualquer cluster de cache autoprojetoado criado será do tipo de nó cache `.r5.large`.

Veja a seguir um exemplo dessa política de permissões ao usar o Valkey ou o RedisOSS.

```
{
  "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"
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

Veja a seguir um exemplo dessa política de permissões ao usar o Memcached.

```

{
  "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 [ElastiCache API|permissões: 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:MaximumDataStorage": "30"
        },
        "StringEquals": {
          "elasticache:DataStorageUnit": "GB"
        }
      }
    }
  ]
}
```

```
}

```

2. `ElastiCache:maximumECPUPerSecond`: especifique o valor máximo 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.

```
{
  "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.

```
{

```

```

"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.

```

{
  "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.

```

{
  "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.

```

{
  "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.

```

{
  "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

```

{
  "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 Valkey ou RedisOSS.

```

{
  "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.

```

{
    "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 que a [CreateReplicationGroup](#) ação especifique que os grupos de replicação só podem ser criados quando TLS não estão sendo usados:

```

{
    "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 não TLS estiver sendo usada (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 os grupos de replicação só podem ser criados quando TLS estão sendo usados:

```

{
  "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. `elasticache:TransitEncryptionEnabled` Defina como `true` para que a `ModifyReplicationGroup` ação especifique que os grupos de replicação só podem ser modificados quando estão TLS sendo usados:

```

{
  "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`. O `TransitEncryptionEnabled` parâmetro definido como `true` também pode ser incluído opcionalmente, mas não é necessário, neste caso, para ser ativado TLS.

`elasticacheAutomaticFailoverEnabled`: especifique que os grupos de replicação seriam criados somente com o failover automático ativado.

```
{
  "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:multiAZEnabled`: especifique que grupos de replicação não podem ser criados com o Multi-AZ desativado.

```
{
  "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.

```
{
  "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 AUTH o token ativado.

```

{
    "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.

```

{
  "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 gerenciadas pelo AWS KMS cliente.

```

{
  "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".

```

{
  "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".

```

{
  "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

```
{
  "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

```

{
  "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

```

{
  "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`.

## Usando funções vinculadas a serviços para a Amazon ElastiCache

A Amazon ElastiCache usa AWS Identity and Access Management (IAM) funções [vinculadas ao serviço](#). Uma função vinculada a serviços é um tipo exclusivo de IAM função vinculada diretamente a um AWS serviço, como a Amazon. ElastiCache As funções ElastiCache vinculadas aos serviços da Amazon são predefinidas pela 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 da Amazon ElastiCache porque você não precisa adicionar manualmente as permissões necessárias. As funções já existem na sua AWS conta, mas estão vinculadas aos casos de ElastiCache uso da Amazon e têm permissões predefinidas. Somente a Amazon ElastiCache pode assumir essas funções, e somente essas funções podem usar a política de permissões predefinida. É possível excluir as funções somente depois de primeiro excluir seus recursos relacionados. Isso protege seus ElastiCache recursos da Amazon porque você não pode remover inadvertidamente as permissões necessárias para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte [AWS Serviços que funcionam com IAM](#) e procure os serviços que têm Sim na coluna Função vinculada ao serviço. Escolha um Sim com um link para visualizar a documentação da função vinculada a esse serviço.

### Sumário

- [Permissões de função vinculadas ao serviço para a Amazon ElastiCache](#)
  - [Permissões para criar uma função vinculada ao serviço](#)

- [Criando uma função vinculada ao serviço \(\) IAM](#)
  - [Criando uma função vinculada ao serviço \(console\) IAM](#)
  - [Criando uma função vinculada ao serviço \(\) IAM CLI](#)
  - [Criando uma função vinculada ao serviço \(\) IAM API](#)
- [Editando a descrição de uma função vinculada ao serviço para a Amazon ElastiCache](#)
  - [Editando uma descrição de função vinculada ao serviço \(console\) IAM](#)
  - [Editando uma descrição de função vinculada ao serviço \(\) IAM CLI](#)
  - [Editando uma descrição de função vinculada ao serviço \(\) IAM API](#)
- [Excluindo uma função vinculada ao serviço para a Amazon ElastiCache](#)
  - [Limpar uma função vinculada ao serviço](#)
  - [Excluindo uma função vinculada ao serviço \(console\) IAM](#)
  - [Excluindo uma função vinculada ao serviço \(\) IAM CLI](#)
  - [Excluindo uma função vinculada ao serviço \(\) IAM API](#)

## Permissões de função vinculadas ao serviço para a Amazon ElastiCache

### Permissões para criar uma função vinculada ao serviço

Para permitir que uma IAM entidade crie uma função AWS ServiceRoleForElastiCache vinculada ao serviço

Adicione a seguinte declaração de política às permissões dessa IAM entidade:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole",
    "iam:PutRolePolicy"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS
ServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

Para permitir que uma IAM entidade exclua a função AWS ServiceRoleForElastiCache vinculada ao serviço

Adicione a seguinte declaração de política às permissões dessa IAM entidade:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS
ServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

Como alternativa, você pode usar uma política AWS gerenciada para fornecer acesso total à Amazon ElastiCache.

Criando uma função vinculada ao serviço () IAM

Você pode criar uma função vinculada ao serviço usando o IAM console, CLI, ou API

Criando uma função vinculada ao serviço (console) IAM

Você pode usar o IAM console para criar uma função vinculada ao serviço.

Para criar uma função vinculada ao serviço (console)

1. Faça login no AWS Management Console e abra o IAM console em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do IAM console, escolha Funções. Em seguida, escolha Criar nova função.
3. Em Selecionar tipo de entidade confiável, selecione Serviço da AWS .
4. Em Ou selecione um serviço para ver seus casos de uso, escolha ElastiCache.
5. Escolha 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.

### Criando uma função vinculada ao serviço () IAM CLI

Você pode usar IAM as operações do AWS Command Line Interface para criar 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
```

### Criando uma função vinculada ao serviço () IAM API

Você pode usar o IAM API para criar 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 o comando [CreateServiceLinkedRole](#) APlligar. Na solicitação, especifique o nome do serviço na forma de `elasticache.amazonaws.com`.

### Editando a descrição de uma função vinculada ao serviço para a Amazon ElastiCache

ElastiCache A Amazon não permite que você edite a função AWS ServiceRoleForElastiCache vinculada ao serviço. 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, você pode editar a descrição da função usando IAM.

### Editando uma descrição de função vinculada ao serviço (console) IAM

Você pode usar o IAM console para editar uma descrição de função vinculada ao serviço.

Para editar a descrição de uma função vinculada ao serviço (console)

1. No painel de navegação do IAM console, escolha Funções.
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.

Editando uma descrição de função vinculada ao serviço () IAM CLI

Você pode usar IAM as operações do AWS Command Line Interface para editar uma descrição de função vinculada ao serviço.

Para alterar a descrição de uma função vinculada ao serviço () CLI

1. (Opcional) Para ver a descrição atual de uma função, use a IAM operação AWS CLI [forget-role](#).

Example

```
$ aws iam get-role --role-name AWS ServiceRoleForElastiCache
```

Use o nome da função, não o ARN, para se referir às funções com as CLI operações. Por exemplo, se uma função tiver o seguinte ARN: `arn:aws:iam::123456789012:role/myrole`, consulte a função como **myrole**.

2. Para atualizar a descrição de uma função vinculada ao serviço, use a operação AWS CLI for IAM. [update-role-description](#)

Para Linux, macOS ou Unix:

```
$ aws iam update-role-description \  
  --role-name AWS ServiceRoleForElastiCache \  
  --description "new description"
```

Para Windows:

```
$ aws iam update-role-description ^\  
  --role-name AWS ServiceRoleForElastiCache ^\  
  --description "new description"
```

Editando uma descrição de função vinculada ao serviço () IAM API

Você pode usar o IAM API para editar uma descrição de função vinculada ao serviço.

Para alterar a descrição de uma função vinculada ao serviço () API

1. (Opcional) Para ver a descrição atual de uma função, use a IAM API operação [GetRole](#).

#### Example

```
https://iam.amazonaws.com/  
?Action=GetRole  
&RoleName=AWS ServiceRoleForElastiCache  
&Version=2010-05-08  
&AUTHPARAMS
```

2. Para atualizar a descrição de uma função, use a IAM API operação [UpdateRoleDescription](#).

#### Example

```
https://iam.amazonaws.com/  
?Action=UpdateRoleDescription  
&RoleName=AWS ServiceRoleForElastiCache  
&Version=2010-05-08  
&Description="New description"
```

## Excluindo uma função vinculada ao serviço para a Amazon ElastiCache

Se você não precisar mais usar um atributo ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. 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.

ElastiCache A Amazon não exclui a função vinculada ao serviço para você.

### Limpar uma função vinculada ao serviço

Antes de poder usar IAM para excluir uma função vinculada ao serviço, primeiro confirme se 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 IAM

1. Faça login no AWS Management Console e abra o IAM console em <https://console.aws.amazon.com/iam/>.

2. No painel de navegação do IAM console, escolha Funções. Em seguida, escolha o nome (não a caixa de seleção) da AWS ServiceRoleForElastiCache função.
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 ElastiCache recursos da Amazon que exigem AWS ServiceRoleForElastiCache

- Para excluir um cluster, consulte o seguinte:
  - [Usando o AWS Management Console](#)
  - [Usando o AWS CLI para excluir um ElastiCache cluster](#)
  - [Usando o ElastiCache API](#)
- Para excluir um grupo de replicação, consulte o seguinte:
  - [Exclusão de um grupo de replicação \(Console\)](#)
  - [Excluir um grupo de replicação \(AWS CLI\)](#)
  - [Excluindo um grupo de replicação \(\) ElastiCache API](#)

Excluindo uma função vinculada ao serviço (console) IAM

Você pode usar o IAM console para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (console)

1. Faça login no AWS Management Console e abra o IAM console em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do IAM console, escolha Funções. 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 Função na parte superior da página, escolha a função Excluir.
4. Na caixa de diálogo de confirmação, revise os últimos dados acessados do serviço, que mostram quando cada uma das funções selecionadas acessou um AWS serviço pela última vez. 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. Assista às notificações do IAM console para monitorar o progresso da exclusão da função vinculada ao serviço. Como a exclusão da função IAM vinculada ao serviço é assíncrona, depois de enviar a função para exclusão, a tarefa de exclusão pode ser bem-sucedida ou falhar. Se a

tarefa obtiver êxito, você poderá escolher Visualizar Detalhes ou Visualizar Recursos a partir das notificações para saber por que a exclusão falhou.

Excluindo uma função vinculada ao serviço () IAM CLI

Você pode usar IAM as operações do AWS Command Line Interface para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada 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 seus nomes de recursos da Amazon (ARNs) em sua conta.

```
$ aws iam get-role --role-name role-name
```

Use o nome da função, não o ARN, para se referir às funções com as CLI operações. Por exemplo, se uma função tem o ARN `arn:aws:iam::123456789012:role/myrole`, você se refere 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.



## Excluindo uma função vinculada ao serviço () IAM API

Você pode usar o IAM API 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 um nome de 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 `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.

## ElastiCache API permissões: referência de ações, recursos e condições

Ao configurar o [controle de acesso](#) e escrever políticas de permissões para anexar a uma IAM política (baseada em identidade ou em recursos), use a tabela a seguir como referência. A tabela lista cada ElastiCache API operação da Amazon e as ações correspondentes para as quais você pode conceder permissões para realizar 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 nenhum recurso ARN, o recurso na política é um curinga (\*).

Você pode usar chaves de condição em suas ElastiCache políticas para expressar condições. Para ver uma lista de chaves ElastiCache de condição específicas, junto com as ações e os tipos de recursos aos quais elas se aplicam, consulte [Uso de chaves de condição](#). Para obter uma lista completa AWS de chaves gerais, consulte as chaves de [contexto de condição AWS global](#) no Guia do IAM usuário.

### Note

Para especificar uma ação, use o `elasticache:` prefixo seguido pelo nome da API operação (por exemplo, `elasticache:DescribeCacheClusters`).

Para ver uma lista de ElastiCache ações, consulte [Ações definidas pela Amazon ElastiCache](#) na Referência de autorização de serviço.

## Validação de conformidade para a Amazon ElastiCache


Audidores terceirizados avaliam a segurança e a AWS conformidade dos AWS serviços como parte de vários programas de conformidade SOC, como PCI, RAMP, Fed 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. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para HIPAA segurança e conformidade na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

 Note

Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a [Referência de serviços HIPAA elegíveis](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização ()). ISO
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#) — Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.

- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

## Mais informações

Para obter informações gerais sobre a conformidade com a AWS nuvem, consulte o seguinte:

- [FIPSEndpoints por serviço](#)
- [Atualizações de serviço em ElastiCache](#)
- [AWS Conformidade na nuvem](#)
- [Modelo de responsabilidade compartilhada](#)
- [AWS PCIDSSPrograma de conformidade](#)

## 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 executem 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](#)
- [Mitigando falhas ao executar 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 autoprotetido, espalhe seus dados em cache por mais nós. Como os clusters autoprotetidos não oferecem suporte para replicação, uma falha de nó sempre resultará em uma certa perda de dados do seu cluster.

Ao criar seu cluster Memcached, você pode criá-lo com 1 a 60 nós, ou mais, mediante 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 autoprotetido, 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 [Regiões e zonas de disponibilidade](#).

## Mitigando falhas ao executar Valkey ou Redis OSS

Ao executar um OSS mecanismo Valkey ou Redis, você tem as seguintes opções para minimizar o impacto de uma falha no nó ou 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. Os clusters autoprotetidos devem ser devidamente configurados para mitigar a falha de um nó individual.

Para mitigar o impacto das falhas dos OSS nós Valkey ou Redis em clusters autoprotetidos, você tem as seguintes opções:

## Tópicos

- [Atenuando falhas: grupos de replicação Valkey ou Redis OSS](#)

### Atenuando falhas: grupos de replicação Valkey ou Redis OSS

Um grupo de OSS replicação Valkey ou Redis é composto por um único nó primário do qual seu aplicativo pode ler e gravar, e de 1 a 5 nós de réplica somente para 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.

### Valkey ou Redis OSS Multi-AZ

Você pode habilitar o Multi-AZ em seus grupos de replicação Valkey ou RedisOSS.

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 [Minimizando o tempo de inatividade 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 autoprotetido, 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 várias 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 os OSS nós Valkey ou Redis, consulte [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#)

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

## Recomendações

É recomendável criar caches sem servidor em clusters autoprojeto, pois você obtém automaticamente uma melhor tolerância a falhas sem configuração adicional. No entanto, durante a criação de um cluster autoprojeto, 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 Valkey ou RedisOSS, 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 autoprojeto no Valkey ou no RedisOSS, recomendamos que você habilite o Multi-AZ em seu grupo de replicação para que ele ElastiCache faça o failover automático para uma réplica 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 RedisOSS, além do acima, recomendamos que você agende 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 em AWS ElastiCache

Como serviço gerenciado, AWS ElastiCache é protegido pelos procedimentos AWS globais de segurança de rede descritos na seção Segurança e Conformidade do [AWS Architecture Center](#).

Você usa API chamadas AWS publicadas para acessar ElastiCache pela rede. Os clientes devem oferecer suporte ao Transport Layer Security (TLS) 1.2 ou posterior. Recomendamos TLS 1.3 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo direto perfeito (), como Ephemeral Diffie-Hellman (PFS) ou Elliptic Curve Ephemeral Diffie-Hellman (). DHE ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. 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 autoprojeto, 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 autoprojeto. Você também pode monitorar o progresso dessas atualizações no ElastiCache cluster selecionado em tempo real.

### Tópicos

- [Gerenciando atualizações de serviços para clusters autoprojeto](#)

## Gerenciando atualizações de serviços para clusters autoprojeto

ElastiCache atualizações de serviço para clusters autoprojeto são lançadas regularmente. Se você tiver um ou mais clusters autoprojeto qualificados para essas atualizações de serviço, receberá notificações por e-mail/SNS, pelo Personal Health Dashboard (PHD) e por CloudWatch eventos da 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 não realizar nenhuma 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 ou AWS CLI aplicar a atualização. API 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 AWS Management Console 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.
  - Data de inscrição recomendada: data de ElastiCache orientação para aplicar as atualizações até.
  - 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
```

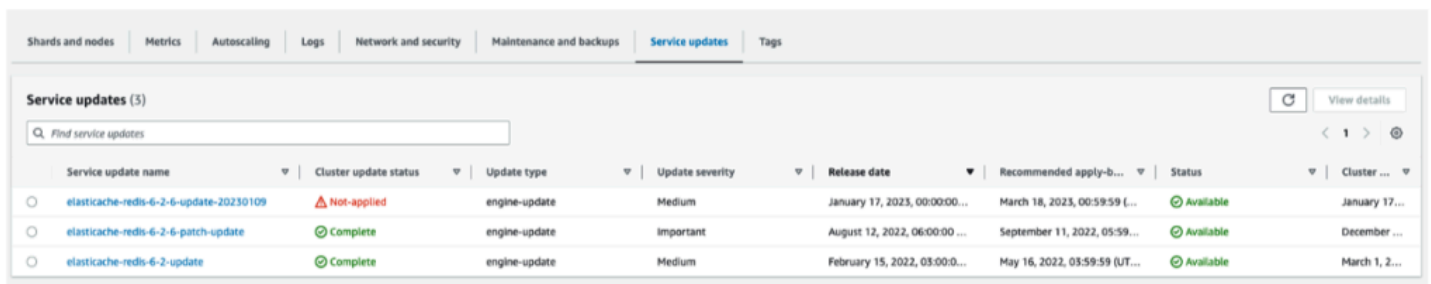
Para obter mais informações, consulte [batch-apply-update-action](#).

## Verificando se você tem a atualização de serviço mais recente aplicada usando o console AWS

Você pode verificar se seus clusters ElastiCache (RedisOSS) estão executando a atualização de serviço mais recente seguindo estas etapas:

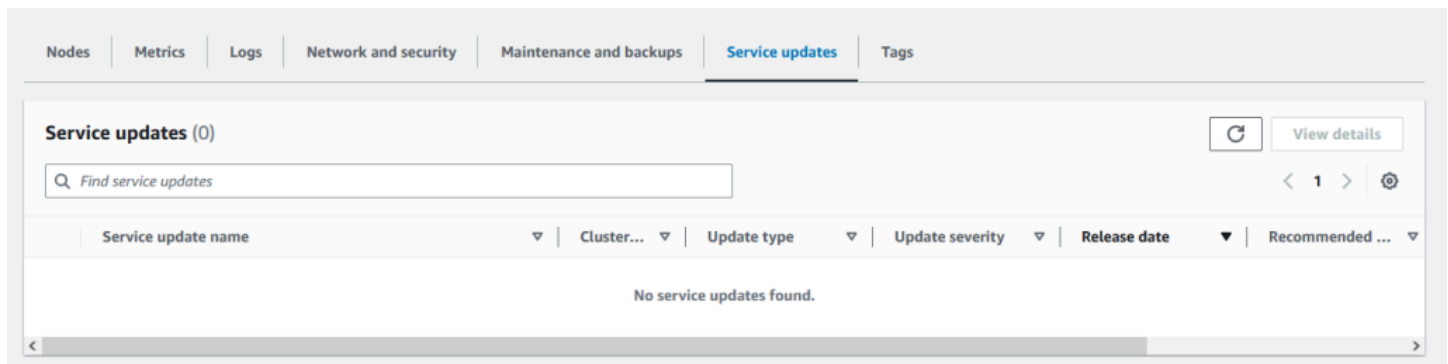
1. Escolha um cluster aplicável na página Redis OSS Clusters
2. Escolha Atualizações de serviço no painel de navegação para ver as atualizações de serviço aplicáveis para esse cluster, se houver.

Se o console exibir uma lista de atualizações de serviço, você poderá selecionar a atualização de serviço e escolher Aplicar agora.



Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-by	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, 05:59:59 (UT...	Available	March 1, 2...

Se o console exibir “Nenhuma atualização de serviço encontrada”, significa que o cluster ElastiCache (RedisOSS) 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 de cache, 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

Vulnerabilidades e exposições comuns (CVE) é uma lista de entradas para vulnerabilidades de segurança cibernética conhecidas publicamente. 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 CVE correções são aplicadas automaticamente ao seu cache. Ao operar clusters autoprotetidos com Valkey ou RedisOSS, ElastiCache expõe o componente. PATCH Por exemplo, ao usar a versão 6.2.6 ElastiCache (RedisOSS), a versão principal é 6, a versão secundária é 2 e a versão do patch é 6. PATCH as versões 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 OSS cluster 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 OSS versão mais recente do ElastiCache Valkey ou do Redis 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. [Gerenciando atualizações de serviços para clusters autoprojeto](#) 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 a CVE for endereçado em uma ElastiCache versão, significa que também é endereçado nas versões mais recentes. Então, por exemplo, se uma vulnerabilidade for abordada na versão 6.0.5 ElastiCache (RedisOSS), 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 ElastiCache (RedisOSS) que executa a versão ElastiCache (RedisOSS) especificada para resolver a vulnerabilidade de segurança. Para obter mais informações sobre como verificar se a atualização de serviço mais recente foi aplicada à versão ElastiCache (RedisOSS) na qual seu cluster está sendo executado, consulte [Gerenciando atualizações de serviços para clusters autoprojeto](#).

ElastiCache Versão (RedisOSS)	CVEsEndereçado
Redis 6.0.5 OSS	<a href="#">CVE-2022-24735</a> *, <a href="#">-2022-24736</a> * <a href="#">CVE</a>
Redis 6.2.6 OSS	<a href="#">CVE-2022-24834</a> *, <a href="#">-2022-35977</a> *, <a href="#">-2022-36021</a> *, <a href="#">CVE-2022-24735</a> , <a href="#">-2022-24736</a> <a href="#">CVE</a> <a href="#">CVE</a>
Redis 7.0.7 OSS	<a href="#">CVE-2023-41056</a> *, <a href="#">CVE -2022-24834</a> *, <a href="#">-2022-35977</a> , <a href="#">-2022-36021</a> , <a href="#">-2022-24735</a> , <a href="#">-2022-24736</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a>

ElastiCache Versão (RedisOSS)	CVEsEndereçado
Redis 7.1.0 OSS	<a href="#">CVE-2023-41056</a> , <a href="#">-2022-24834</a> , <a href="#">-2022-35977</a> , <a href="#">CVE-2022-36021</a> , <a href="#">-2022-24735</a> , <a href="#">-2022-24736</a> <a href="#">CVE CVE CVE CVE</a>

# Registro e monitoramento na Amazon ElastiCache

Para gerenciar o cache, é importante que você saiba o desempenho dos caches. ElastiCache gera métricas que são publicadas no Amazon CloudWatch Logs para monitorar o desempenho do seu cache. Além disso, ElastiCache gera eventos quando mudanças significativas acontecem em seus recursos de cache (por exemplo, um novo cache é criado ou um cache é excluído).

## Tópicos

- [Métricas e eventos sem servidor para Valkey e Redis OSS](#)
- [Métricas e eventos sem servidor para Memcached](#)
- [Registrando ElastiCache API chamadas da Amazon com AWS CloudTrail](#)
- [SNSMonitoramento de ElastiCache eventos pela Amazon](#)
- [Entrega de logs](#)
- [Monitorando o uso com CloudWatch métricas](#)
  
- [Registrando ElastiCache API chamadas da Amazon com AWS CloudTrail](#)

## Métricas e eventos 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 e eventos de cluster autoprotetidos para Valkey e Redis OSS](#)

## Métricas de cache sem servidor

O AWS/ElastiCache namespace inclui as seguintes CloudWatch métricas para seus caches sem servidor Valkey ou RedisOSS.

## Códigos métricos para Valkey ou Redis OSS

Métrica	Descrição	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
CurrConnections	O número de conexões cliente com o cache.	Contagem
ThrottledCmds	O número de solicitações que foram limitadas ElastiCache porque a carga de trabalho	Contagem

Métrica	Descrição	Unidade
	estava aumentando mais rápido do que pode ser escalada. ElastiCache	
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 comTTL.	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
IamAuthenticationExpirations	O número total de conexões Valkey ou Redis IAM autenticadas e expiradas. OSS Você pode encontrar mais informações sobre o <a href="#">Autenticação com IAM</a> no guia do usuário.	Contagem

Métrica	Descrição	Unidade
IamAuthenticationThrottling	O número total de solicitações ou solicitações Valkey ou Redis IAM autenticadas por limitação. OSS AUTH HELLO Você pode encontrar mais informações sobre o <a href="#">Autenticação com IAM</a> 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 malsucedidas de autenticação no Valkey ou no Redis OSS usando o comando. AUTH Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
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	Descrição	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 de estatísticas de OSS comando Valkey ou Redis. Ele é derivado somando todos o tipos de comandos geo: geoadd, geodist, geohash, geopos, georadius e georadiusbymember.	Contagem
GeoSpatialBasedCmdsECPUs	ECPUs consumido por comandos baseados em localização geoespacial.	Contagem
GetTypeCmds	O número total de comandos tipo somente leitura. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos do tipo somente leitura (get, hget, scard, lrange e assim por diante).	Contagem

Métrica	Descrição	Unidade
GetTypeCmdsECPUs	ECPUs consumido por comandos de leitura.	Contagem
HashBasedCmds	O número total de comandos baseados em hash. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que atuam em um ou mais hashes (hget, hkeys, hvals, hdel e assim por diante).	Contagem
HashBasedCmdsECPUs	ECPUs consumido por comandos baseados em hash.	Contagem
HyperLogLogBasedCmds	O número total de comandos HyperLogLog baseados. Isso é derivado da estatística de OSS commandstats do Valkey ou do Redis pela soma de todos os comandos do tipo pf (pfadd, pfcount, pfmerge e assim por diante).	Contagem
HyperLogLogBasedCmdsECPUs	ECPUs consumido por comandos HyperLogLog baseados.	Contagem



Métrica	Descrição	Unidade
JsonBasedCmds	O número total de JSON comandos, incluindo comandos de leitura e gravação. Isso é derivado da OSS estatística de comandos Valkey ou Redis, somando todos os JSON comandos que atuam sobre as teclas. JSON	Contagem
JsonBasedCmdsECPUs	ECPUsconsumido por todos os JSON comandos, incluindo comandos de leitura e gravação.	Contagem
JsonBasedGetCmds	O número total de comandos JSON somente para leitura. Isso é derivado da OSS estatística de comandos Valkey ou Redis, somando todos os comandos de JSON leitura que atuam sobre as chaves. JSON	Contagem
JsonBasedGetCmdsECPUs	ECPUsconsumido por comandos JSON somente de leitura.	Contagem
JsonBasedSetCmds	O número total de comandos de JSON gravação. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos de JSON gravação que atuam sobre as chaves. JSON	Contagem

Métrica	Descrição	Unidade
JsonBasedSetCmdsECPUs	ECPUs consumido por comandos de JSON gravação.	Contagem
KeyBasedCmds	O número total de comandos baseados em chave. Isso é derivado da OSS estatística de comandos Valkey ou Redis, somando todos os comandos que atuam em uma ou mais chaves em várias estruturas de dados (del, expire, renomeie etc.).	Contagem
KeyBasedCmdsECPUs	ECPUs consumido por comandos baseados em teclas.	Contagem
ListBasedCmds	O número total de comandos baseados em lista. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que atuam em uma ou mais listas (lindex, lrange, lpush, ltrim e assim por diante).	Contagem
ListBasedCmdsECPUs	ECPUs consumido por comandos baseados em listas.	Contagem

Métrica	Descrição	Unidade
NonKeyTypeCmds	O número total de comandos não baseados em chave. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que não atuam em uma chave, por exemplo, acl, dbsize ou info.	Contagem
NonKeyTypeCmdsECPUs	ECPUsconsumido por non-key-based comandos.	Contagem
PubSubBasedCmds	O número total de comandos para a funcionalidade pub/sub. Isso é derivado das OSS estatísticas de comandos Valkey ou Redis, somando todos os comandos usados para a funcionalidade pub/sub: psubscribe, publish, pubsub, punsubscribe, ssubscribe e, sunsubscribe, spublish, subscribe e unsubscribe.	Contagem
PubSubBasedCmdsECPUs	ECPUsconsumido por comandos baseados em pub/sub.	Contagem

Métrica	Descrição	Unidade
SetBasedCmds	O número total de comandos que são baseados em conjuntos. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que atuam em um ou mais conjuntos (scard, sdiff, sadd, union e assim por diante).	Contagem
SetBasedCmdsECPUs	ECPUsconsumido por comandos baseados em conjuntos.	Contagem
SetTypeCmds	O número total de tipos de comando de gravação. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os tipos mutativos de comandos que operam nos dados (set, hset, sadd, lpop e assim por diante).	Contagem
SetTypeCmdsECPUs	ECPUsconsumido por comandos de gravação.	Contagem

Métrica	Descrição	Unidade
SortedSetBasedCmds	O número total de comandos que são classificados com base em conjuntos. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que atuam em um ou mais conjuntos ordenados (zcount, zrange, zrank, zadd e assim por diante).	Contagem
SortedSetBasedCmdsECPUs	ECPUsconsumido por comandos baseados em classificação.	Contagem
StringBasedCmds	O número total de comandos baseados em string. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que atuam em uma ou mais strings (strlen, setex, setrange e assim por diante).	Contagem
StringBasedCmdsECPUs	ECPUsconsumido por comandos baseados em strings.	Contagem

Métrica	Descrição	Unidade
StreamBasedCmds	O número total de comandos que são baseados em fluxo. Isso é derivado da OSS estatística de comandos Valkey ou Redis pela soma de todos os comandos que atuam em um ou mais tipos de dados de streams (xrange, xlen, xadd, xdel e assim por diante).	Contagem
StreamBasedCmdsECPUs	ECPUsconsumido 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 `ElastiCache API DescribeEvents`.

Você pode escolher monitorar, ingerir, transformar e agir em ElastiCache eventos usando a Amazon EventBridge. Saiba mais no Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

### Visualizando ElastiCache eventos (Console)

Para visualizar eventos usando o ElastiCache console:

1. Faça login no AWS Management Console 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, a GMT hora 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	Descrição	Unidade	Origem	Message
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

Detail-Type	Descrição	Unidade	Origem	Message
				cache 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 VPC endpoint.
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 criar VPC endpoint.
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 VPC endpoint.



Detail-Type	Descrição	Unidade	Origem	Message
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Um usuário com uma OSS versão incompatível do Valkey ou do Redis 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	Descrição	Unidade	Origem	Message
Falha na criação de cache	Fio de cache	falha	serverless-cache	<p>Falha na criação do cache &lt;cache-name&gt;. Houve falha na restauração de dados a partir do snapshot porque &lt;reason&gt;.</p> <p>Motivos de falha:</p> <ul style="list-style-type: none"> <li>• falhou na recuperação do arquivo do S3.</li> <li>• O md5 esperado não corresponde ao md5 real.</li> <li>• o RDB arquivo fornecido tem uma versão não suportada.</li> </ul>

### Eventos de atualização de cache sem servidor (Valkey ou Redis) OSS

Detail-Type	Lista de recursos	Categoria	Origem	Message
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	SecurityGroups atualizado para o cache<cache-name>.

Detail-Type	Lista de recursos	Categoria	Origem	Message
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 OSS versão incompatível do Valkey ou do Redis 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	Origem	Message
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	Falha na atualização do cache <cache-name>. 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	Origem	Message
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	Descrição	Unidade	Origem	Message
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	Descrição	Unidade	Origem	Message
				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 instantâneo de cache sem servidor (Valkey ou Redis) OSS

Detail-Type	Resources-list	Categoria	Origem	Message
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	<p>Falha na criação do snapshot para o cache &lt;cache-name&gt;.</p> <p>Falha na criação do snapshot &lt;snapshot-name&gt; com a chave gerenciada pelo cliente &lt;key-id&gt; &lt;reason&gt;.</p> <p>Mensagens de motivo da falha:</p> <ul style="list-style-type: none"> <li>• porque a chave gerenciada pelo cliente está desabilitada</li> <li>• porque não foi possível encontrar a chave gerenciada pelo cliente</li> </ul>

Detail-Type	Resources-list	Categoria	Origem	Message
				<ul style="list-style-type: none"> <li>porque a solicitação atingiu o tempo limite</li> </ul>
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	<p>Falha na criação do snapshot para o cache &lt;cache-name&gt;. Houve falha na criação do snapshot &lt;snapshot-name&gt; por causa de &lt;reason&gt;.</p> <p>Motivo padrão:</p> <ul style="list-style-type: none"> <li>por causa de um erro interno</li> </ul>
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	<p>Falha na exportação do snapshot para o cache &lt;cache-name&gt;. Não foi possível exportar o instantâneo para o bucket %s porque ElastiCache não tem permissões para o bucket.</p>

Detail-Type	Resources-list	Categoria	Origem	Message
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	Origem	Message
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	Origem	Message
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	Origem	Message
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 autoprotetidos para Valkey e Redis OSS

ElastiCache oferece uma ampla variedade de métricas e eventos para monitorar clusters autoprotetidos ao trabalhar com Valkey e Redis. OSS Isso inclui métricas em nível de host, métricas em nível de comando e registros de eventos disponíveis por meio do AWS CLI e do Amazon Simple Notification Service (SNS).

### Tópicos

- [Métricas para clusters autoprojeto](#)
- [Eventos para clusters autoprojeto \(Valkey e Redis\) OSS](#)

## Métricas para clusters autoprojeto

Quando você projeta clusters por conta própria, 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 Valkey e Redis OSS](#).

## Eventos para clusters autoprojeto (Valkey e Redis) OSS

ElastiCache registra eventos relacionados aos seus caches projetados por você mesmo. Ao trabalhar com clusters autoprojeto, 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 autoprojeto não são publicados na Amazon EventBridge.

As informações do evento de cluster autoprojeto 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 `ElastiCache API 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 AWS Management Console 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, a GMT hora 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 autoprotetidos.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

O código a seguir lista todos os eventos de caches autoprotetidos nas últimas 24 horas (1.440 minutos).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

### Eventos de cluster autoprotetidos


Esta seção contém a lista de eventos que devem ser recebidos para os clusters autoprotetidos.

Os ElastiCache eventos a seguir acionam SNS as notificações da Amazon. Para obter informações sobre detalhes de eventos, consulte [Visualizando ElastiCache eventos](#).


Nome do evento	Message	Descrição
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Um nó de cache foi adicionado ao cluster de cache 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 de cache foram alterados.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioning	O provisionamento de um cluster de cache está

Nome do evento	Message	Descrição
	Complete <i>cluster-name-0001-005</i>	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 iniciar um novo cluster de cache em uma nuvem privada virtual inexistente (VPC).
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	Message	Descrição
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>Um dos seguintes eventos ocorreu:</p> <ul style="list-style-type: none"><li>• A lista de security groups de cache autorizados para o cluster de cache foi modificada.</li><li>• Um ou mais novos grupos EC2 de segurança foram autorizados em qualquer um dos grupos de segurança de cache associados ao cluster de cache.</li><li>• Um ou mais grupos EC2 de segurança foram revogados de qualquer um dos grupos de segurança de cache associados ao cluster de cache.</li></ul>

Nome do evento	Message	Descrição
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e começou a substituir o nó de cache.</p> <div data-bbox="1068 493 1507 758" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>A DNS entrada para o nó de cache substituído não foi alterada.</p></div> <p>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.</p>



Nome do evento	Message	Descrição
ElastiCache:CacheNodesReplaceComplete	ElastiCache:CacheNodesReplaceComplete : <i>cluster-name</i>	<p>ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e concluiu a substituição do nó de cache.</p> <div data-bbox="1068 495 1507 758" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>A DNS entrada para o nó de cache substituído não foi alterada.</p> </div> <p>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.</p>
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>Um ou mais nós de cache foram reinicializados.</p> <p>Mensagem (Memcached): "Cache node %s shutdown" Em seguida, uma segunda mensagem: "Cache node %s restarted"</p>

Nome do evento	Message	Descrição
ElastiCache: CertificateRenewalComplete (Somente Valkey ou RedisOSS)	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 de cache e todos os nós de cache associados foi concluída.
ElastiCache:FailoverComplete (Somente Valkey ou RedisOSS)	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.
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.

Nome do evento	Message	Descrição
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 <a href="#">Substituindo nós (Valkey e RedisOSS)</a> .
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 <a href="#">Substituindo nós (Valkey e RedisOSS)</a> .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Um nó de cache foi removido do cluster de cache.
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.

Nome do evento	Message	Descrição
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Uma atualização de autoatendimento está disponível para o nó.
ElastiCache: SnapshotComplete (Somente Valkey ou RedisOSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Um snapshot de cache foi concluído com sucesso.
ElastiCache: SnapshotFailed (Somente Valkey ou RedisOSS)	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 <a href="#">DescribeSnapshots</a> , o status será failed.

## Métricas e eventos sem servidor para Memcached

Esta seção descreve as métricas e os eventos que você pode monitorar ao trabalhar com caches sem servidor.

### Tópicos

- [Métricas de cache sem servidor](#)
- [Eventos de cache sem servidor](#)

## Métricas de cache sem servidor

Esta seção descreve as métricas e os eventos que você pode monitorar ao trabalhar com caches sem servidor do Memcached.

O AWS/ElastiCache namespace inclui as seguintes CloudWatch métricas para seus caches sem servidor do Memcached.

Métrica	Descrição	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	Descrição	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	Descrição	Unidade
CmdGet	O número de comandos get recebidos pelo cache.	Contagem

Métrica	Descrição	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	Descrição	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	Descrição	Unidade
CmdFlush	O número de comandos flush recebidos pelo cache.	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 `ElastiCache API DescribeEvents`.

Você pode escolher monitorar, ingerir, transformar e agir em ElastiCache eventos usando a Amazon EventBridge. Saiba mais no Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

### Visualizando ElastiCache eventos (Console)

Para visualizar eventos usando o ElastiCache console:

1. Faça login no AWS Management Console 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, a GMT hora 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	Descrição	Unidade	Origem	Message
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 VPC endpoint.
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

Detail-Type	Descrição	Unidade	Origem	Message
				cota atingido para criar VPC endpoint.
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 VPC endpoint.

#### Eventos de atualização do cache sem servidor (Memcached)

Detail-Type	Lista de recursos	Categoria	Origem	Message
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.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	<cache-name>Falha na atualização do

Detail-Type	Lista de recursos	Categoria	Origem	Message
				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 eles são inválidos.

#### Eventos de exclusão de cache sem servidor (Memcached)

Detail-Type	Lista de recursos	Categoria	Origem	Message
Cache excluído	Fio de cache	exclusão	serverless-cache	O cache <cache-name> foi excluído.

#### Eventos de limite de uso de cache sem servidor (Memcached)

Detail-Type	Descrição	Unidade	Origem	Message
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	Limites atualizados para o cache <cache-name>.

Detail-Type	Descrição	Unidade	Origem	Message
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 instantâneo de cache sem servidor (Memcached)

Detail-Type	Resources-list	Categoria	Origem	Message
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	Origem	Message
				<p>a pelo cliente &lt;key-id&gt; &lt;reason&gt;.</p> <p>Mensagens de motivo da falha:</p> <ul style="list-style-type: none"><li>• porque a chave gerenciada pelo cliente está desabilitada</li><li>• porque não foi possível encontrar a chave gerenciada pelo cliente</li><li>• porque a solicitação atingiu o tempo limite</li></ul>

Detail-Type	Resources-list	Categoria	Origem	Message
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	<p>Falha na criação do snapshot para o cache &lt;cache-name&gt;. Houve falha na criação do snapshot &lt;snapshot-name&gt; por causa de &lt;reason&gt;.</p> <p>Motivo padrão:</p> <ul style="list-style-type: none"><li>• por causa de um erro interno</li></ul>
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	<p>Falha na exportação do snapshot para o cache &lt;cache-name&gt;. Não foi possível exportar o instantâneo para o bucket %s porque ElastiCache não tem permissões para o bucket.</p>

Detail-Type	Resources-list	Categoria	Origem	Message
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	Origem	Message
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	Origem	Message
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	Origem	Message
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'.

## Registrando ElastiCache API chamadas da Amazon com AWS CloudTrail

ElastiCache A Amazon está integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço na Amazon ElastiCache. CloudTrail captura todas as API chamadas para a Amazon ElastiCache como eventos, incluindo chamadas do ElastiCache console da Amazon e de chamadas de código para ElastiCache API as operações da

Amazon. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para a Amazon ElastiCache. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita à Amazon ElastiCache, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

## ElastiCache Informações da Amazon em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre na Amazon ElastiCache, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo de eventos em sua AWS conta, incluindo eventos para a Amazon ElastiCache, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- [Visão Geral para Criar uma Trilha](#)
- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando as SNS notificações da Amazon para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

Todas as ElastiCache ações da Amazon são registradas CloudTrail e documentadas na [ElastiCache APIReferência](#). Por exemplo, chamadas para o `CreateCacheCluster`, `DescribeCacheCluster` e `ModifyCacheCluster` as ações geram entradas nos arquivos de CloudTrail log.

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 raiz ou de IAM usuário.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o [CloudTrail userIdentityElemento](#).

## Entendendo as entradas do arquivo de ElastiCache log da Amazon

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das API chamadas públicas, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateCacheCluster` ação.

```
{
  "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 CloudTrail registro que demonstra a DescribeCacheCluster ação. Observe que para todas as chamadas do Amazon ElastiCache Describe\* Describe (), a ResponseElements seção 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 CloudTrail registro que registra uma `ModifyCacheCluster` ação.

```

{
  "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"
  }
}
```

## SNSMonitoramento de ElastiCache eventos pela Amazon

Quando eventos significativos acontecem em um cluster, ElastiCache envia uma notificação para um SNS tópico específico da Amazon. 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 SNS notificações ElastiCache da Amazon](#)
- [Visualizando ElastiCache eventos](#)
- [Notificações de eventos e Amazon SNS](#)

## Gerenciando SNS notificações ElastiCache da Amazon

Você pode configurar ElastiCache para enviar notificações para eventos importantes do cluster usando o Amazon Simple Notification Service (AmazonSNS). Nesses exemplos, você configurará um cluster com o Amazon Resource Name (ARN) de um SNS tópico da Amazon para receber notificações.

### Note

- Este tópico pressupõe que você se inscreveu na Amazon SNS e configurou e se inscreveu em um tópico da AmazonSNS. Para obter informações sobre como fazer isso, consulte o [Guia do desenvolvedor do Amazon Simple Notification Service](#).
- Por padrão, API `modify-replication-group` isso 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 `--notification-topic-arn` opção de criar um tópico separado para esse grupo.

## Adicionar um SNS tópico da Amazon

As seções a seguir mostram como adicionar um SNS tópico da Amazon usando o AWS Console AWS CLI, o ou ElastiCache API o.

### Adicionar um SNS tópico da Amazon (Console)

O procedimento a seguir mostra como adicionar um SNS tópico da Amazon a um cluster. Ao usar o Valkey ou o Redis OSS para adicionar um SNS tópico da Amazon 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 SNS tópico da Amazon.

Para adicionar ou modificar um SNS tópico da Amazon para um cluster (console)

1. Faça login no AWS Management Console e abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. Em Clusters, escolha o cluster ao qual você deseja adicionar ou modificar um SNS tópico da AmazonARN.
3. Escolha Modificar.
4. Em Modificar cluster, em Tópico para SNS notificação, escolha o SNS tópico que você deseja adicionar ou escolha ARNEntrada manual e digite o SNS tópico ARN da Amazon.
5. Escolha Modificar.

Adicionando um SNS tópico da Amazon (AWS CLI)

Para adicionar ou modificar um SNS tópico da Amazon para um cluster, use o AWS CLI comandomodify-cache-cluster.

O exemplo de código a seguir adiciona um SNS tópico arn da Amazon ao 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](#).

## Adicionando um SNS tópico da Amazon (ElastiCache API)

Para adicionar ou modificar um SNS tópico da Amazon para um cluster, chame a `ModifyCacheCluster` ação 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](#).

## Ativando e desativando as notificações da Amazon SNS

Você pode ativar ou desativar notificações para um cluster. Os procedimentos a seguir mostram como desativar as SNS notificações da Amazon.

## Ativando e desativando SNS as notificações da Amazon (console)

Para desativar as SNS notificações da Amazon usando o AWS Management Console

1. Faça login no AWS Management Console 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 seus clusters executando Valkey ou RedisOSS, no painel de navegação, escolha 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 Modificar cluster, em Tópico para SNS notificação, escolha Desativar notificações.
6. Escolha Modificar.

## Ativando e desativando SNS as notificações da Amazon (AWS CLI)

Para desativar SNS as notificações da Amazon, 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
```

## Ativando e desativando SNS as notificações da Amazon (ElastiCache API)

Para desativar SNS as notificações da Amazon, chame a `ModifyCacheCluster` ação 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 facilmente recuperar eventos do registro usando o ElastiCache console, o AWS CLI `describe-events` comando ou a ElastiCache API ação `DescribeEvents`.

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 AWS Management Console 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 GMT hora 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 de cache.

```
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 cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:51:14.095Z"
    }
  ]
}
```

```
    },
    {
      "SourceIdentifier": "mycluster-0003-004",
      "SourceType": "cache-cluster",
      "Message": "Cache cluster created",
      "Date": "2020-06-09T01:51:14.094Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-005",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0001 in availability zone us-west-2b",
      "Date": "2020-06-09T01:42:55.603Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-005",
      "SourceType": "cache-cluster",
      "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:42:55.576Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-005",
      "SourceType": "cache-cluster",
      "Message": "Cache cluster created",
      "Date": "2020-06-09T01:42:55.574Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-004",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0001 in availability zone us-west-2b",
      "Date": "2020-06-09T01:28:40.798Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-004",
      "SourceType": "cache-cluster",
      "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:28:40.775Z"
    },
    {
      "SourceIdentifier": "mycluster-0001-004",
      "SourceType": "cache-cluster",
      "Message": "Cache cluster created",
      "Date": "2020-06-09T01:28:40.773Z"
    }
  ]
}
```



```
    }  
  ]  
}
```

Para obter mais informações, como os parâmetros disponíveis e os valores de parâmetros permitidos, consulte [describe-events](#).

## Visualizando ElastiCache eventos (ElastiCache API)

Para gerar uma lista de ElastiCache eventos usando o 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 Amazon SNS

ElastiCache pode publicar mensagens usando o Amazon Simple Notification Service (SNS) quando eventos significativos acontecem em um cluster de cache. 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 SNS documentação da Amazon, consulte a [página SNS do produto Amazon](#).

As notificações são publicadas em um SNS tópico específico da Amazon. Os seguintes são requisitos para notificações:

- Somente um tópico pode ser configurado para ElastiCache notificações.
- A AWS conta proprietária do SNS tópico da Amazon deve ser a mesma conta que possui o cluster de cache no qual as notificações estão habilitadas.

- O SNS tópico da Amazon para o qual você está publicando não pode ser criptografado.

### Note

É possível anexar um SNS tópico criptografado (em repouso) da Amazon 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 SNS tópico da Amazon precisa estar na mesma região do ElastiCache cluster.


## ElastiCache Eventos

Os ElastiCache eventos a seguir acionam SNS as notificações da Amazon. Para obter informações sobre detalhes de eventos, consulte [Visualizando ElastiCache eventos](#).

Nome do evento	Message	Descrição
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Um nó de cache foi adicionado ao cluster de cache 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 de cache foram alterados.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	O provisionamento de um cluster de cache está concluído, e os nós de cache no cluster de cache estão prontos para uso.
ElastiCache: CacheClusterProvisioningFailed devido	ElastiCache:CacheClusterProvisioning	Foi feita uma tentativa de iniciar um novo cluster de

Nome do evento	Message	Descrição
ao estado de rede incompatível	Failed : <i>cluster-name</i>	cache em uma nuvem privada virtual inexistente (VPC).
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: <ul style="list-style-type: none"> <li>A lista de security groups de cache autorizados para o cluster de cache foi modificada.</li> <li>Um ou mais novos grupos EC2 de segurança foram autorizados em qualquer um dos grupos de segurança de cache associados ao cluster de cache.</li> <li>Um ou mais grupos EC2 de segurança foram revogados de qualquer um dos grupos de segurança de cache associados ao cluster de cache.</li> </ul>

Nome do evento	Message	Descrição
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e começou a substituir o nó de cache.</p> <div data-bbox="1068 495 1507 758"><p> <b>Note</b></p><p>A DNS entrada para o nó de cache substituído não foi alterada.</p></div> <p>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.</p>

Nome do evento	Message	Descrição
ElastiCache:CacheNodesReplaceComplete	ElastiCache:CacheNodesReplaceComplete : <i>cluster-name</i>	<p>ElastiCache detectou que o host que executa um nó de cache está degradado ou inacessível e concluiu a substituição do nó de cache.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>A DNS entrada para o nó de cache substituído não foi alterada.</p> </div> <p>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.</p>
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>Um ou mais nós de cache foram reinicializados.</p> <p>Mensagem (Memcached): "Cache node %s shutdown" Em seguida, uma segunda mensagem: "Cache node %s restarted"</p>

Nome do evento	Message	Descrição
ElastiCache: CertificateRenewalComplete (Somente Valkey ou RedisOSS)	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 de cache e todos os nós de cache associados foi concluída.
ElastiCache:FailoverComplete (Somente Valkey ou RedisOSS)	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.
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.

Nome do evento	Message	Descrição
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 <a href="#">Substituindo nós (Valkey e RedisOSS)</a> .
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 <a href="#">Substituindo nós (Valkey e RedisOSS)</a> .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Um nó de cache foi removido do cluster de cache.
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.



Nome do evento	Message	Descrição
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Uma atualização de autoatendimento está disponível para o nó.
ElastiCache: SnapshotComplete (Somente Valkey ou RedisOSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Um snapshot de cache foi concluído com sucesso.
ElastiCache: SnapshotFailed (Somente Valkey ou RedisOSS)	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 <a href="#">DescribeSnapshots</a> , o status será failed.

## Tópicos relacionados

- [Visualizando ElastiCache eventos](#)

## Entrega de logs

### Note

O Slow Log é compatível com Valkey 7.x e versões posteriores, e clusters de OSS cache e grupos de replicação do Redis usando a versão 6.0 do motor em diante.

O Engine Log é compatível com Valkey 7.x e clusters de OSS cache e grupos de replicação do Redis usando a versão 6.2 do mecanismo em diante.

A entrega de registros permite que você transmita [SLOWLOG](#)o Engine Log para um dos 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 registro será entregue ao destino especificado em um dos dois formatos: JSON ou TEXT.

Um número fixo de entradas de registro lento é recuperado do mecanismo periodicamente. 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 Slow Log contém as seguintes informações:

- `CacheClusterId`— O ID do cluster de cache
- `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 de cache
- CacheNodeId— O ID do nó de cache
- Nível de registro — LogLevel pode ser um dos seguintes:VERBOSE("-"),NOTICE("\*"),WARNING("#").
- Hora — A UTC hora 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 o processo filho do escritor trabalhando RDB em/ AOF ou "X" para sentinela.
- Mensagem — Mensagem de registro do mecanismo.

## Permissões para configurar os logs

Você precisa incluir as seguintes IAM permissões em sua política de IAM usuário/função:

- 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 tanto JSON quanto TEXT

Veja a seguir um exemplo de JSON formato:

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

Veja a seguir um exemplo de TEXT formato:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

### Log do mecanismo

O registro do motor suporta tanto JSON quanto TEXT

Veja a seguir um exemplo de JSON formato:

```
{
  "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"
}
```

Veja a seguir um exemplo de TEXT formato:

```
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 10000 milliseconds.
```

## 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 registros de vários OSS clusters e grupos de replicação do Valkey ou do Redis podem ser entregues ao mesmo grupo de registros.
- Um novo fluxo de log será criado para cada nó dentro de um cluster de cache 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:

```
{
  "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 stream de entrega do Firehose no qual os registros serão entregues.
- Os registros de vários OSS clusters e grupos de replicação do Valkey ou do Redis podem ser entregues ao mesmo stream de entrega.
- Os logs de cada nó dentro de um cluster de cache 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 registros para Firehose não está disponível na região Ásia-Pacífico (Osaka).

## Permissões para publicar registros 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingFHSLR",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "ElastiCacheLoggingFH",
      "Action": [
        "firehose:TagDeliveryStream"
      ],
      "Resource": "Amazon Kinesis Data Firehose delivery stream ARN",
      "Effect": "Allow"
    }
  ]
}
```

## Especificação da entrega de logs usando o console

Usando o, AWS Management Console você pode criar um cluster Valkey ou Redis OSS (modo de cluster desativado) seguindo as etapas em [Criação de um cluster Valkey \(modo de cluster desativado\) \(console\)](#) ou criar um cluster Valkey ou Redis OSS (modo de cluster ativado) usando as etapas em. [Criação de um cluster Valkey ou Redis OSS \(modo de cluster ativado\) \(console\)](#) Em ambos os casos, você configura a entrega de logs fazendo o seguinte:

1. Em Configurações avançadas, escolha Registros e, em seguida, marque Registros lentos ou Registros do mecanismo.
2. Em Formato de registro, escolha Texto ou JSON.
3. Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
4. Em Destino do log, escolha Criar novo e insira o nome do bucket do Amazon S3, o nome do grupo de CloudWatchLogs log ou o nome do stream do Kinesis Data Firehose, ou escolha Selecionar existente e, em seguida, escolha o nome do grupo Logs ou CloudWatch o nome do stream 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 ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha clusters Valkey ou clusters 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 Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.



- Em Destino do registro, escolha Criar novo e insira o nome do seu grupo de CloudWatchLogs registros ou o nome do stream do Kinesis Data Firehose. Ou escolha Selecionar existente e, em seguida, escolha o nome do seu grupo de CloudWatchLogs registros ou o nome do stream do Kinesis Data Firehose.

## Especificando a entrega de registros usando o AWS CLI

### Log lento

Crie um grupo de replicação com entrega lenta de CloudWatch registros para o 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 fornecer registros lentos aos CloudWatch registros

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 a entrega de registros 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"
}'
```

Modifique um grupo de replicação para entregar o log do mecanismo ao 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, CPU uso) 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 de cache começar a diminuir. Para obter mais informações, consulte [Que métricas devo monitorar?](#) neste guia.

## Tópicos

- [Métricas em nível de host](#)

- [Métricas para 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 Valkey e Redis OSS](#)

Métrica	Descrição	Unidade
CPUUtilization	A porcentagem de CPU utilização de todo o host. Como o Valkey e o Redis OSS são de um único segmento, recomendamos que você monitore a EngineCPUUtilization métrica para nós com 4 ou mais. vCPUs	Percentual
CPUCreditBalance	<p>O número de CPU créditos ganhos que uma instância acumulou desde que foi lançada ou iniciada. Para o T2 Standard, isso CPUCredit Balance também inclui o número de créditos de lançamento que foram acumulados.</p> <p>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,</p>	Créditos (v CPU - minutos)

Métrica	Descrição	Unidade
	<p>todos os novos créditos ganhos serão descartados. Para a T2 Padrão, os créditos de execução não são contabilizados para o limite.</p> <p>Os créditos contidos no CPUCreditBalance estão disponíveis para que a instância gaste além de sua utilização básicaCPU.</p> <p>CPUs métricas de crédito estão disponíveis apenas com uma frequência de cinco minutos.</p> <p>Essa métrica não está disponível para instâncias expansíveis T2.</p>	
CPUCreditUsage	<p>O número de CPU créditos gastos pela instância para CPU utilização. Um CPU crédito é igual a um v CPU funcionando com 100% de utilização por um minuto ou uma combinação equivalente de vCPUs utilização e tempo (por exemplo, um v CPU funcionando com 50% de utilização por dois minutos ou dois rodando com 25% de utilização por dois minutos vCPUs ).</p> <p>CPUs métricas de crédito estão disponíveis apenas com uma frequência de cinco minutos. Se você especificar um período de mais cinco minutos, use a estatística Sum (Soma) em vez da estatística Average (Média).</p> <p>Essa métrica não está disponível para instâncias expansíveis T2.</p>	Créditos (v CPU - minutos)



Métrica	Descrição	Unidade
FreeableMemory	A quantidade de memória livre disponível no host. Isso é derivado dos RAM buffers e do cache que o sistema operacional relata como liberáveis.	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
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

Métrica	Descrição	Unidade
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 explosão máxima por segundo de bytes recebidos em cada minuto.	Bytes
NetworkMaxBytesOut	A explosão máxima por segundo de bytes transmitidos em cada minuto.	Bytes
NetworkMaxPacketsIn	A explosão máxima por segundo de pacotes recebidos em cada minuto.	Contagem
NetworkMaxPacketsOut	A explosão máxima por segundo de pacotes transmitidos em cada minuto.	Contagem
SwapUsage	A quantidade de troca usada no host.	Bytes

## Métricas para Valkey e Redis OSS

O Amazon ElastiCache namespace inclui as seguintes métricas do Valkey e do Redis. OSS Essas métricas são as mesmas ao usar o mecanismo Valkey.

Com exceção de ReplicationLag e EngineCPUUtilization, essas métricas são derivadas do info comando. 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	Descrição	Unidade
ActiveDefragHits	O número de realocações de valor por minuto executada pelo processo de desfragme	Número

Métrica	Descrição	Unidade
	ntação ativo. Isso é derivado da <code>active_defrag_hits</code> estatística em <a href="#">INFO</a> .	
AuthenticationFailures	O número total de tentativas malsucedidas de autenticação no Valkey ou no Redis OSS usando o comando. AUTH Você pode encontrar mais informações sobre falhas de autenticação individuais usando o <a href="#">ACLLLOG</a> comando. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
	O número total de bytes alocados pelo Valkey ou pelo Redis OSS para todos os fins, incluindo o conjunto de dados, os buffers e assim por diante.	Bytes
BytesUsedForCache	Dimension: Tier=Memory para OSS clusters Valkey ou Redis usando <a href="#">Hierarquização de dados em ElastiCache</a> : O número total de bytes usados para cache por memória. Esse é o valor da <code>used_memory</code> estatística em <a href="#">INFO</a> .	Bytes
	Dimension: Tier=SSD para OSS clusters Valkey ou Redis usando <a href="#">Hierarquização de dados em ElastiCache</a> : 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 <a href="#">Hierarquização de dados em ElastiCache</a> .	Bytes


Métrica	Descrição	Unidade
BytesWrittenToDisk	O número total de bytes gravados no disco por minuto. Compatível somente para clusters usando <a href="#">Hierarquização de dados em ElastiCache</a> .	Bytes
CacheHits	O número de buscas de chaves somente leitura bem-sucedidas no dicionário principal. Isso é derivado da <code>keyspace_hits</code> estatística em <a href="#">INFO</a> .	Contagem
CacheMisses	O número de buscas de chaves somente leitura malsucedidas no dicionário principal. Isso é derivado da <code>keyspace_misses</code> estatística em <a href="#">INFO</a> .	Contagem
CommandAuthorizationFailures	O número total de tentativas falhadas por usuários para executar comandos que eles não têm permissão para chamar. Você pode encontrar mais informações sobre falhas de autenticação individuais usando o <a href="#">ACLLOG</a> comando. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
CacheHitRate	Indica a eficiência de uso da instância Valkey ou RedisOSS. 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

Métrica	Descrição	Unidade
ChannelAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar canais que eles não têm permissão para acessar. Você pode encontrar mais informações sobre falhas de autenticação individuais usando o <a href="#">ACLLOG</a> comando. Sugerimos definir um alarme nesta métrica para detectar tentativas de acesso não autorizadas.	Contagem
CurrConnections	O número de conexões de clientes, excluindo conexões de réplicas de leitura. ElastiCache usa de duas a quatro conexões para monitorar o cluster em cada caso. Isso é derivado da <code>connected_clients</code> estatística em <a href="#">INFO</a> .	Contagem
CurrItems	O número de itens no cache. Isso é derivado da <code>keyspace</code> estatística, somando todas as chaves em todo o espaço de teclas.	Contagem
	Dimension: Tier=Memory para clusters usando <a href="#">Hierarquização de dados em ElastiCache</a> . O número de itens em memória.	Contagem
	Dimension: Tier=SSD (unidades de estado sólido) para clusters usando <a href="#">Hierarquização de dados em ElastiCache</a> . O número de itens em SSD.	Contagem
CurrVolatileItems	Número total de chaves em todos os bancos de dados que têm um ttl definido. Isso é derivado da <code>expires</code> estatística, somando todas as chaves com um ttl definido em todo o espaço de teclas.	Contagem

Métrica	Descrição	Unidade
DatabaseCapacityUsagePercentage	<p>Porcentagem da capacidade total de dados para o cluster que está em uso.</p> <p>Em instâncias com camadas de dados, a métrica é calculada como <math>(\text{used\_memory} - \text{mem\_not\_counted\_for\_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})</math>, de onde <code>used_memory</code> e de onde <code>maxmemory</code> são <a href="#">INFO</a>tiradas.</p> <p>Em todos os outros casos, a métrica é calculada usando <math>\text{used\_memory} / \text{maxmemory}</math>.</p>	Percentual
DatabaseCapacityUsageCountedForEvictPercentage	<p>Porcentagem da capacidade total de dados do cluster que está em uso, excluindo a memória usada para sobrecarga e. COB Essa métrica é calculada como:</p> $\frac{\text{used\_memory} - \text{mem\_not\_counted\_for\_evict}}{\text{maxmemory}}$ <p>Em instâncias de dados em camadas, a métrica é calculada como:</p> $\frac{(\text{used\_memory} + \text{SSD used})}{(\text{maxmemory} + \text{SSD total capacity})}$ <p>de onde <code>used_memory</code> e de onde <code>maxmemory</code> são retirados <a href="#">INFO</a>.</p>	Percentual
DatabaseMemoryUsagePercentage	<p>Porcentagem de memória para o cluster que está em uso. Isso é calculado usando <math>\text{used\_memory} / \text{maxmemory}</math> from <a href="#">INFO</a>.</p>	Percentual

Métrica	Descrição	Unidade
DatabaseMemoryUsageCountedForEvictPercentage	Porcentagem da memória do cluster que está em uso, excluindo a memória usada para sobrecarga e. COB Isso é calculado usando <code>used_memory-mem_not_counted_for_evict/maxmemory</code> from <a href="#">INFO</a> .	Percentual
DB0AverageTTL	Exposições <code>avg_ttl</code> DBO da keyspace estatística de comando. <a href="#">INFO</a> 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 expulsa por causa delaLRU), ele sintetiza um DEL comando, que é transmitido para todas as réplicas. Portanto, DB0Average TTL é 0 para nós de réplica, devido ao fato de que eles não expiram as chaves e, portanto, não rastreiam TTL.	Milissegundos

Métrica	Descrição	Unidade
EngineCPUUtilization	<p>Fornece CPU a utilização do encadeamento do mecanismo Valkey ou RedisOSS. Como o Valkey e o Redis OSS são de um único segmento, você pode usar essa métrica para analisar a carga do processo em si. A EngineCPUUtilization métrica fornece uma visibilidade mais precisa do processo. Você pode usá-lo em conjunto com a CPUUtilization métrica. CPUUtilization expõe CPU a utilização da instância do servidor como um todo, incluindo outros sistemas operacionais e processos de gerenciamento. Para tipos de nós maiores com quatro vCPUs ou mais, use a EngineCPUUtilization métrica para monitorar e definir limites para escalabilidade.</p>	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 CPU carga de trabalho. Isso não é significativo em hosts maiores com mais de dois vCPUs. Mas isso pode afetar hospedeiros menores com 2 vCPUs ou menos. Se você monitorar apenas a EngineCPUUtilization métrica, não terá conhecimento das situações em que o host está sobrecarregado com o alto CPU uso do Valkey ou do Redis OSS e com o alto CPU uso




Métrica	Descrição	Unidade
	<p>dos processos de monitoramento em segundo plano. Portanto, recomendamos monitorar a <code>CPUUtilization</code> métrica para hosts com dois vCPUs ou menos.</p>	
Evictions	O número de chaves que foram removidas devido ao limite <code>maxmemory</code> . Isso é derivado da <code>evicted_keys</code> estatística em <a href="#">INFO</a> .	Contagem
GlobalDatastoreReplicationLag	Esse é o atraso entre o nó primário da região secundária e o nó primário da região primária. Para Valkey ou Redis habilitado para o modo de <code>clusterOSS</code> , o atraso indica o atraso máximo entre os fragmentos.	Segundos
IamAuthenticationExpirations	O número total de conexões Valkey ou Redis IAM autenticadas e expiradas. OSS Você pode encontrar mais informações sobre o <a href="#">Autenticação com IAM</a> no guia do usuário.	Contagem
IamAuthenticationThrottling	O número total de solicitações ou solicitações Valkey ou Redis IAM autenticadas por limitação. OSS AUTH HELLO Você pode encontrar mais informações sobre o <a href="#">Autenticação com IAM</a> 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	Descrição	Unidade
KeyAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar chaves que eles não têm permissão para acessar. Você pode encontrar mais informações sobre falhas de autenticação individuais usando o <a href="#">ACLLOG</a> comando. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
KeysTracked	O número de chaves que estão sendo rastreadas pelo rastreamento de chaves Valkey ou Redis OSS como uma porcentagem 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 motor Valkey ou RedisOSS. Certos limites significarão comportamentos diferentes. O valor recomendado é ter fragmentação acima de 1,0. Isso é calculado a partir <code>mem_fragmentation_ratio</code> statistic de <a href="#">INFO</a> .	Número

Métrica	Descrição	Unidade
NewConnections	<p>O número total de conexões que foram aceitas pelo servidor durante esse período. Isso é derivado da <code>total_connections_received</code> estatística em <a href="#">INFO</a>.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se você estiver usando ElastiCache (RedisOSS) versão 5 ou inferior, entre duas e quatro das conexões relacionadas por essa métrica serão usadas ElastiCache para monitorar o cluster. No entanto, ao usar a versão 6 ou superior do ElastiCache (RedisOSS), as conexões usadas ElastiCache para monitorar o cluster não são incluídas nessa métrica.</p> </div>	Contagem
NumItemsReadFromDisk	O número total de itens recuperados do disco por minuto. Compatível somente para clusters usando <a href="#">Hierarquização de dados em ElastiCache</a> .	Contagem
NumItemsWrittenToDisk	O número total de itens gravados no disco por minuto. Compatível somente para clusters usando <a href="#">Hierarquização de dados em ElastiCache</a> .	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 ativadoOSS. EC2 O valor de 1 indica que os dados não estão sincronizados. Para concluir a migração, use a <a href="#">CompleteMigrationAPI</a> operação.	Booleano

Métrica	Descrição	Unidade
Reclaimed	O número total de eventos de expiração de chaves. Isso é derivado da <code>expired_keys</code> estatística em <a href="#">INFO</a> .	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. Isso é derivado da <code>master_repl_offset</code> estatística em <a href="#">INFO</a> .	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 Valkey 7.2 e versões posteriores e do OSS motor Redis 5.0.6 em diante, 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. Isso é derivado da <code>rdb_bgsave_in_progress</code> estatística em <a href="#">INFO</a> .	Booleano

Métrica	Descrição	Unidade
TrafficManagementActive	<p>Indica se o ElastiCache (RedisOSS) 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 pelo 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.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>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 .</p> </div>	Booleano

## EngineCPUUtilization disponibilidade

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 (N. da Virgínia)
us-west-1	Oeste dos EUA (N. da Califórnia)

Região	Nome da região
us-west-2	Oeste dos EUA (Oregon)
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 (Barém)
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)

Região	Nome da região
sa-east-1	América do Sul (São Paulo)
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 com base no tipo de comando, incluindo o número de chamadas, o CPU tempo total consumido por esses comandos e a média CPU 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 usando 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 CPU o tempo gasto ElastiCache para processar o comando. Observe que, para clusters que usam camadas de dados, o tempo necessário para buscar itens não SSD está incluído nessas medições.

Para obter uma lista completa dos comandos disponíveis, consulte [os comandos](#) na documentação do Valkey.

Métrica	Descrição	Unidade
ClusterBasedCmds	O número total de comandos que são baseados em cluster. Isso é derivado da commandstats estatística pela soma de todos os comandos que atuam em um cluster (cluster slot,cluster info, e assim por diante).	Contagem
ClusterBasedCmdsLatency	Latência de comandos baseados em cluster.	Microsegundos
EvalBasedCmds	O número total de comandos para comandos baseados em avaliação. Isso é derivado da	Contagem

Métrica	Descrição	Unidade
	commandstats estatística pela soma eval de,. evalsha	
EvalBasedCmdsLatency	Latência de comandos baseados em avaliação.	Microssegundos
GeoSpatialBasedCmds	O número total de comandos para comandos baseados em dados geoespaciais. Isso é derivado da commandstats estatística. Ele é derivado somando todos o tipos de comandos geo: geoadd, geodist, geohash, geopos, georadius, e georadiusbymember.	Contagem
GeoSpatialBasedCmdsLatency	Latência de comandos baseados em dados geoespaciais.	Microssegundos
GetTypeCmds	O número total de comandos do tipo read-only . Isso é derivado da commandstats estatística pela soma de todos os comandos de read-only tipo (get,, hget scardlrange, e assim por diante.)	Contagem
GetTypeCmdsLatency	Latência de comandos de leitura.	Microssegundos
HashBasedCmds	O número total de comandos baseados em hash. Isso é derivado da commandstats estatística pela soma de todos os comandos que atuam em um ou mais hashes (hget,, hkeys hvalshdel, e assim por diante).	Contagem
HashBasedCmdsLatency	Latência de comandos baseados em hash.	Microssegundos



Métrica	Descrição	Unidade
HyperLogLogBasedCmds	O número total de comandos baseados em HyperLogLog . Isso é derivado da <code>commandstats</code> estatística pela soma de todos os pf tipos de comandos (pfadd,pfcount,pfmerge, etc.).	Contagem
HyperLogLogBasedCmdsLatency	Latência dos comandos HyperLogLog baseados.	Microssegundos
JsonBasedCmds	O número total de JSON comandos, incluindo comandos de leitura e gravação. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os JSON comandos que atuam sobre JSON as teclas.	Contagem
JsonBasedCmdsLatency	Latência de todos os JSON comandos, incluindo comandos de leitura e gravação.	Microssegundos
JsonBasedGetCmds	O número total de comandos JSON somente para leitura. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos de JSON leitura que atuam sobre JSON as teclas.	Contagem
JsonBasedGetCmdsLatency	Latência de comandos somente JSON para leitura.	Microssegundos
JsonBasedSetCmds	O número total de comandos de JSON gravação. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos de JSON gravação que atuam sobre JSON as chaves.	Contagem
JsonBasedSetCmdsLatency	Latência dos comandos de JSON gravação.	Microssegundos

Métrica	Descrição	Unidade
KeyBasedCmds	O número total de comandos baseados em chave. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que atuam em uma ou mais chaves em várias estruturas de dados ( <code>del</code> , <code>expire</code> , <code>rename</code> , e assim por diante.).	Contagem
KeyBasedCmdsLatency	Latência de comandos baseados em chave.	Microssegundos
ListBasedCmds	O número total de comandos baseados em lista. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que atuam em uma ou mais listas ( <code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> , e assim por diante).	Contagem
ListBasedCmdsLatency	Latência de comandos baseados em lista.	Microssegundos
NonKeyTypeCmds	O número total de comandos não baseados em chave. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que não atuam em uma tecla, 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 a funcionalidade pub/sub. Isso é derivado das <code>commandstats</code> estatísticas pela soma de todos os comandos usados para a funcionalidade do pub/sub: <code>psubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>unsubscribe</code> , e <code>spublish</code> , <code>subscribe</code> , <code>unsubscribe</code> .	Contagem

Métrica	Descrição	Unidade
PubSubBasedCmdsLatency	Latência de comandos baseados em pub/sub.	Microssegundos
SetBasedCmds	O número total de comandos que são baseados em conjuntos. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que atuam em um ou mais conjuntos ( <code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>sunion</code> , e assim por diante).	Contagem
SetBasedCmdsLatency	Latência de comandos baseados em conjunto.	Microssegundos
SetTypeCmds	O número total de tipos de comando write. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os mutative tipos de comandos que operam nos dados ( <code>set</code> , <code>hset</code> , <code>sadd</code> , <code>lpop</code> , e assim por diante).	Contagem
SetTypeCmdsLatency	Latência de comandos de gravação.	Microssegundos
SortedSetBasedCmds	O número total de comandos que são classificados com base em conjuntos. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que atuam em um ou mais conjuntos ordenados ( <code>zcount</code> , <code>zrange</code> , <code>zrank</code> , <code>zadd</code> , e assim por diante).	Contagem
SortedSetBasedCmdsLatency	Latência de comandos baseados em classificação.	Microssegundos

Métrica	Descrição	Unidade
StringBasedCmds	O número total de comandos baseados em string. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que atuam em uma ou mais strings ( <code>strlen</code> , <code>setexstrange</code> , e assim por diante).	Contagem
StringBasedCmdsLatency	Latência de comandos baseados em string.	Microssegundos
StreamBasedCmds	O número total de comandos que são baseados em fluxo. Isso é derivado da <code>commandstats</code> estatística pela soma de todos os comandos que atuam em um ou mais tipos de dados de fluxos ( <code>xrange</code> , <code>xlen</code> , <code>xaddxdel</code> , e assim por diante).	Contagem
StreamBasedCmdsLatency	Latência de comandos baseados em fluxo.	Microssegundos

## Métricas para o Memcached

O namespace `AWS/ElastiCache` inclui as métricas do Memcached a seguir.

O `ElastiCache` namespace `AWS` inclui as seguintes métricas derivadas do comando de estatísticas 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

Métrica	Descrição	Unidade
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 CAS (verificadas e definidas) que o cache recebeu 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

Métrica	Descrição	Unidade
<code>CurrConnections</code>	<p>A contagem do número de conexões do cache em determinado momento. ElastiCache usa duas a três conexões para monitorar o cluster.</p> <p>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.</p> <p>O total de conexões é a soma das conexões de cliente, as conexões para monitoramento e as conexões internas mencionadas acima.</p>	Contagem
<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

Métrica	Descrição	Unidade
Evictions	O número de itens não expirados removidos pelo cache para permitir espaço para novas gravações.	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
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

Métrica	Descrição	Unidade
CmdTouch	O número cumulativo de solicitações touch.	Contagem
CurrConfig	O número atual de configurações armazenadas.	Contagem
EvictedUnfetched	O número de itens válidos despejados do cache (LRU) usado menos recentemente que nunca foram tocados após serem configurados.	Contagem
ExpiredUnfetched	O número de itens expirados recuperados dos LRU quais nunca foram tocados após serem configurados.	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 ElastiCache namespace AWS/inclui as seguintes métricas calculadas em nível de cache.

Métrica	Descrição	Unidade
NewConnections	O número de novas conexões recebidas pelo cache. Isso é derivado da estatística total_connections do Memcached registrando a alteração em total_connections durante um período. Isso sempre será pelo menos 1, devido a uma conexão reservada para ElastiCache a.	Contagem
NewItems	O número de novos itens armazenados pelo cache. Isso é derivado da estatística total_ite	Contagem



Métrica	Descrição	Unidade
	ms do Memcached registrando a alteração em total_items ao longo de um período.	
UnusedMemory	<p>A quantidade de memória não usada pelos dados. Isso é derivado das estatísticas limit_maxbytes e bytes do Memcached subtraindo bytes de limit_maxbytes.</p> <p>Como a sobrecarga do Memcached usa memória além da usada pelos dados, não UnusedMemory deve ser considerada a quantidade de memória disponível para dados adicionais. Você pode passar por remoções, mesmo que você ainda tenha alguma memória não utilizada.</p> <p>Para obter informações mais detalhadas, consulte <a href="#">Memcached item memory usage</a>.</p>	Bytes

## 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](#)
- [EngineCPUUtilization](#)
- [SwapUsage \(Valkey e RedisOSS\)](#)
- [Evictions](#)
- [CurrConnections](#)
- [Memória \(Valkey e RedisOSS\)](#)
- [Rede](#)
- [Latência](#)
- [Replicação](#)
- [Gerenciamento de tráfego \(Valkey e RedisOSS\)](#)

### 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 2 vCPUs ou menos, use a `CPUUtilization` métrica para monitorar sua carga de trabalho.

De um modo geral, sugerimos que você defina seu limite em 90% do que está disponível CPU. Como o Valkey e o Redis OSS são ambos de thread único, o valor limite real 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 proveniente de solicitações de leitura, amplie seu cluster de cache adicionando réplicas de leitura. Se a workload

principal é de solicitações de gravação, dependendo da configuração do cluster, recomendamos que você:

- Clusters Valkey ou Redis OSS (modo de cluster desativado): amplie usando um tipo de instância de cache maior.
- Clusters Valkey ou Redis OSS (modo de cluster ativado): adicione mais fragmentos para distribuir a carga de trabalho de gravação em mais nós primários.

#### Tip

Em vez de usar a métrica `Host-LevelCPUUtilization`, OSS os usuários do Valkey e do Redis podem usar a métrica `EngineCPUUtilization`, que relata a porcentagem de uso no núcleo do mecanismo Valkey ou Redis. OSS Para ver se essa métrica está disponível em seus nós e para obter mais informações, consulte [Métricas para Valkey e Redis OSS](#).

Para tipos de nós maiores com 4 vCPUs ou mais, talvez você queira usar a `EngineCPUUtilization` métrica, que relata a porcentagem de uso no núcleo do OSS mecanismo Valkey ou Redis. Para ver se essa métrica está disponível em seus nós e para obter mais informações, consulte [Métricas para Redis OSS](#).

## Memcached

Como o Memcached é multi-threaded, essa métrica pode atingir até 90%. Se você exceder esse limite, aumente seu cluster de cache usando um tipo de nó de cache maior ou escale para baixo adicionando mais nós de cache.

## EngineCPUUtilization

Para tipos de nós maiores com 4 vCPUs ou mais, talvez você queira usar a `EngineCPUUtilization` métrica, que relata a porcentagem de uso no núcleo do OSS mecanismo Redis. Para ver se essa métrica está disponível em seus nós e para obter mais informações, consulte [Métricas para Valkey e Redis OSS](#).

Para obter mais informações, consulte a CPU seção [Monitoramento das melhores práticas com a Amazon ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#).

## SwapUsage (Valkey e RedisOSS)

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:

- [Garantindo que você tenha memória suficiente para criar um instantâneo do Valkey ou do Redis OSS](#)
- [Gerenciando a 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 seu cluster usando um tipo de nó maior ou expanda 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 a Amazon ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#).

## Memória (Valkey e RedisOSS)

A memória é um aspecto central do Valkey e do RedisOSS. 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. As estatísticas sobre a utilização da memória de um nó estão disponíveis na seção de memória do [INFO](#) comando.

Para obter mais informações, consulte a seção Memória em [Monitoramento das melhores práticas com a Amazon ElastiCache \(RedisOSS\) 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 a Amazon ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#).

## Latência

Você pode medir a latência de um comando com um conjunto de CloudWatch métricas que fornecem latências agregadas por estrutura de dados. Essas métricas de latência são calculadas usando a `commandstats` estatística do comando `INFOValkey`.

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 a Amazon ElastiCache \(RedisOSS\) usando a Amazon CloudWatch](#).

## Gerenciamento de tráfego (Valkey e RedisOSS)

ElastiCache (RedisOSS) gerencia automaticamente o tráfego em relação a um nó quando mais comandos de entrada são enviados ao nó do que os que podem ser processados pelo Valkey ou pelo Redis. 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 CloudWatch permita que você escolha qualquer estatística e período para cada métrica, nem todas as combinações serão úteis. Por exemplo, as estatísticas Média, Mínimo e Máximo de CPUUtilization são úteis, mas a estatística de Soma não.

Todas as ElastiCache amostras são publicadas por um período 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 de cache 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.



```
--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 o CloudWatch API

Para coletar estatísticas CPU de utilização de um cluster de cache

- Chame o 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=mycachecluster,CacheNodeId=0002`

Example



```
http://monitoring.amazonaws.com/  
  ?Action=GetMetricStatistics  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &StartTime=2018-07-05T00:00:00  
  &EndTime=2018-07-06T23:59:00  
  &Period=3600  
  &Statistics.member.1=Average  
  &Dimensions.member.1="CacheClusterId=mycachecluster"  
  &Dimensions.member.2="CacheNodeId=0002"  
  &Namespace=&AWS;/ElastiCache  
  &MeasureName=CPUUtilization  
  &Timestamp=2018-07-07T17%3A48%3A21.746Z  
  &AWS;AccessKeyId=<&AWS; Access Key ID>  
  &Signature=<Signature>
```

# Cotas para ElastiCache

Sua AWS conta tem cotas padrão, anteriormente chamadas de limites, para cada AWS serviço. A menos que especificado de outra forma, cada cota é específica da região . É possível solicitar aumentos para algumas cotas e outras cotas não podem ser aumentadas.

Para ver as cotas de ElastiCache, abra o console [Service Quotas](#). No painel de navegação, escolha AWS serviços e selecione ElastiCache.

Para solicitar o aumento da cota, consulte [Solicitar um 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 AWS conta tem as seguintes cotas relacionadas a. ElastiCache

Recurso	Padrão
Caches sem servidor por região	40
Instantâneos sem servidor por dia por cache, Redis	24
Nós por região	300
Nós por cluster, Memcached	60
Nodes por cluster por tipo de instância, Valkey ou Redis OSS (modo de cluster ativado)	90
Nodes por fragmento, Valkey ou Redis OSS (modo de cluster desativado)	6
Grupos de parâmetros por região	300
Grupos de segurança por região	50
Grupos de sub-redes por região	300
Sub-redes por grupo de sub-redes	20

Recurso	Padrão
Usuários por grupo de usuários, Redis	100
Número máximo de usuários, Redis	1000
Número máximo de grupos de usuários, Redis	100

# Referência

Os tópicos desta seção abrangem o trabalho com a Amazon ElastiCache API e a ElastiCache seção do AWS CLI. Também estão incluídas nesta seção mensagens de erro comuns e notificações de serviço.

- [Usando o ElastiCache API](#)
- [ElastiCache APIReferência](#)
- [ElastiCache seção da AWS CLI Referência](#)
- [Mensagens ElastiCache de erro da Amazon](#)
- [Notificações](#)

## Usando o ElastiCache API

Esta seção fornece descrições orientadas a tarefas de como usar e implementar ElastiCache operações. Para obter uma descrição completa dessas operações, consulte a [ElastiCache APIReferência da Amazon](#).

### Tópicos

- [Usando a consulta API](#)
- [Bibliotecas disponíveis](#)
- [Solução de problemas de aplicações](#)

## Usando a consulta API

### Parâmetros de consulta

HTTPSolicitações baseadas em consultas são HTTP solicitações que usam o HTTP verbo 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`. Valores de `n` são números inteiros a partir de 1.

## Autenticação de solicitação de consulta

Você só pode enviar solicitações de consulta HTTPS e deve incluir uma assinatura em cada solicitação 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 pressupõe que você esteja registrado AWS e tenha uma ID de chave de acesso e uma chave de acesso secreta.

### Processo de autenticação de consulta

1. O remetente constrói uma solicitação para AWS
2. O remetente calcula a assinatura da solicitação, um código de autenticação de mensagens baseado em hash (HMAC) com uma função 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) para AWS
4. AWS usa o ID da chave de acesso para pesquisar a chave de acesso secreta.
5. AWS gera uma assinatura a partir dos dados da solicitação e da chave de acesso secreta usando o mesmo algoritmo usado para calcular a assinatura 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 da string de consulta UTF -8 pelo nome do parâmetro com a ordem natural de bytes. Os parâmetros podem vir do GET URI ou do POST corpo (quando Content-Type é x-www-form-urlencoded application/).
  - b. URLcodifique o nome e os valores do parâmetro de acordo com as seguintes regras:
    - i. Não URL codifique nenhum dos caracteres não reservados definidos por 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. Codificação percentual estendida UTF de -8 caracteres no formato %XY%ZA...
    - iv. Codificar em percentual o caractere de espaço como %20 (e não +, como em esquemas de codificação comuns).
  - c. Separe os nomes dos parâmetros codificados de seus valores codificados com o sinal de igual (=) (ASCIIcaractere 61), mesmo que o valor do parâmetro esteja vazio.
  - d. Separe os pares nome-valor com um e comercial (&) (ASCIIfórmula 38).
2. Crie a string a ser assinada de acordo com a seguinte pseudogramática (o "\n" representa uma nova linha). ASCII

```
StringToSign = HTTPVerb + "\n" +  
ValueOfHostHeaderInLowercase + "\n" +  
HTTPRequestURI + "\n" +  
CanonicalizedQueryString <from the preceding step>
```

O HTTPRequestURI componente é o componente do caminho HTTP absoluto URI até, mas não inclui, a string de consulta. Se HTTPRequestURI estiver vazio, use uma barra (/).

3. Calcule um RFC compatível HMAC com 2104 com a string que você acabou de criar, sua chave de acesso secreta como chave SHA256 e/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
```

Para a sequência de caracteres de consulta anterior, você calcularia a HMAC assinatura sobre a sequência de caracteres a seguir.

```
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=2877960fced9040b41b4feaca835fd5cf9264f768e6a0236c9143f915ffa56
```

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

AWS fornece kits de desenvolvimento de software (SDKs) para desenvolvedores de software que preferem criar aplicativos usando linguagens específicas APIs em vez do Query. API Eles SDKs fornecem funções básicas (não incluídas no APIs), como autenticação de solicitações, novas tentativas de solicitação e tratamento de erros, para facilitar o início. SDKse recursos adicionais estão disponíveis para as seguintes linguagens de programação:

- [Java](#)
- [Windows e .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Para obter informações sobre outras linguagens, consulte [Código e bibliotecas de exemplo](#).

## Solução de problemas de aplicações

ElastiCache fornece erros específicos e descritivos para ajudá-lo a solucionar problemas ao interagir com o. ElastiCache API

### 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 um `Error` nó na resposta do ElastiCache API.

XPathA sintaxe fornece uma maneira simples de pesquisar a presença de um `Error` nó, bem como uma maneira fácil de recuperar o código e a mensagem de erro. O trecho de código a seguir usa Perl



e o XPath móduloXML:: 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 o. ElastiCache API

- Verifique se ElastiCache está funcionando corretamente.

Para fazer isso, basta abrir uma janela do navegador e enviar uma solicitação de consulta ao ElastiCache serviço (como <https://elasticsearch.amazonaws.com>). Um erro interno do servidor `MissingAuthenticationTokenException` ou 500 confirma que o serviço está disponível e respondendo às solicitações.

- Verificação da estrutura de sua solicitação.

Cada ElastiCache operação tem uma página de referência na ElastiCache APIReferência. 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.

ElastiCache tem um fórum de discussão onde você pode procurar soluções para problemas que outras pessoas enfrentaram ao longo do caminho. Para exibir o fórum, consulte

<https://forums.aws.amazon.com/> .

## Configurando a interface da linha de ElastiCache comando

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 deste tópico somente se você estiver indo AWS CLI para ElastiCache.

### Important

A Amazon ElastiCache Command Line Interface (CLI) não oferece suporte a nenhuma ElastiCache melhoria após a API versão 2014-09-30. Para usar a ElastiCache funcionalidade mais recente da linha de comando, use a [interface da linha de AWS comando](#).

## 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 possa trabalhar em um ambiente Linux/ UNIX ou Windows. As ferramentas de linha de ElastiCache comando da Amazon também funcionam no Mac OS X, que é um ambiente UNIX baseado; no entanto, nenhuma instrução específica do Mac OS X está incluída 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/e **C:\>** para um comando UNIX 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 JDK instalação JRE ou outra é aceitável. Para visualizar e baixar JREs para uma variedade de plataformas, incluindo Linux/ UNIX e Windows, consulte Downloads do [Java SE](#).

## 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 como o caminho completo do diretório que contém um subdiretório chamado bin que, por sua vez, contém o executável java (no Linux eUNIX) ou java.exe (no Windows) executável.

Para configurar da variável Java Home

### 1. Configure a variável Java Home.

- No Linux eUNIX, digite o seguinte comando:

```
$ 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 de linha de comando estão disponíveis como um ZIP arquivo no [site ElastiCache Developer Tools](#). Essas ferramentas são escritas em Java e incluem scripts de shell para Windows 2000/XP/Vista/Windows 7, Linux/ e Mac. UNIX OSX O ZIP arquivo é independente e não é necessária nenhuma instalação; basta baixar o arquivo zip e descompactá-lo em um diretório em sua máquina local.

## Configuração das ferramentas

As ferramentas da linha de comando dependem de uma variável de ambiente (AWS\_ELASTICACHE\_HOME) para localizar as bibliotecas de suporte. 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 denominado ElastiCacheCli - A.b.nnnn (A, B e n são números de versão/lançamento) e contém subdiretórios chamados bin e lib.

Para definir a variável de HOME ambiente AWS ELASTICACHE \_\_

- Abra uma janela de linha de comando e insira um dos comandos a seguir para definir a variável de HOME ambiente AWS ELASTICACHE \_\_.
- No Linux eUNIX, digite o seguinte comando:

```
$ 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, recomendamos que você adicione o BIN diretório das ferramentas ao seu sistemaPATH. O restante deste guia pressupõe que o BIN diretório esteja no caminho do sistema.

Para adicionar o BIN diretório das ferramentas ao caminho do sistema

- Digite os comandos a seguir para adicionar o BIN diretório das ferramentas ao seu sistemaPATH.
- No Linux eUNIX, digite o seguinte comando:

```
$ 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 AWS Acesso e da Chave de Acesso Secreta fornecidas com sua AWS conta. 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 editar com suas informações. A seguir encontra-se o conteúdo do arquivo de modelo:

```
AWS AccessKeyId=<Write your AWS access ID>  
AWS SecretKey=<Write your AWS secret key>
```

### Important

UNIXAtivado, limite as permissões ao proprietário do arquivo de credencial:

```
$ chmod 600 <the file created above>
```

Com a configuração do arquivo de credenciais, você precisará definir a variável de FILE ambiente `AWS_CREDENTIAL__` para que as ElastiCache ferramentas possam encontrar suas informações.

Para definir a variável de FILE ambiente `AWS_CREDENTIAL__`

1. Defina a variável de ambiente:

- No Linux eUNIX, 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 ver a página de uso de todos os ElastiCache comandos.

## Variáveis de ambiente

As variáveis de ambiente podem ser úteis para scripting, configuração de padrões ou para substituí-los temporariamente.

Além da variável de FILE ambiente `AWS_CREDENTIAL__`, a maioria das API ferramentas incluídas na interface de linha de ElastiCache comando suporta as seguintes variáveis:

- `EC2_REGION` — A AWS região a ser usada.
- `AWS_ELASTICACHE_URL` — O URL a ser usado para a chamada de serviço. Não é necessário especificar um endpoint regional diferente se `EC2_REGION` for especificado ou se o parâmetro `--region` for passado.

Os exemplos a seguir mostram como definir a variável ambiental EC2 \_ REGION para configurar a região usada pelas API ferramentas:

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, Redis ou OSS Memcached. 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 cache 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 abrange ElastiCache notificações nas quais você pode 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.

As notificações não anunciam recursos ou ElastiCache funcionalidades novos ou aprimorados.

### ElastiCache Notificações gerais

Atualmente, não há ElastiCache notificações pendentes que não sejam específicas do mecanismo.


### ElastiCache Notificações (Memcached)

As ElastiCache notificações a seguir são específicas do mecanismo Memcached.

#### ElastiCache Notificações específicas (Memcached)

- [Alerta: o rastreador Memcached está causando LRU falhas de segmentação](#)

#### Alerta: o rastreador Memcached está causando LRU falhas de segmentação

 Data do alerta: 28 de fevereiro de 2017

Em algumas circunstâncias, seu cluster pode apresentar instabilidade com uma falha de segmentação no LRU Memcached Crawler. Este é um problema no mecanismo Memcached que existe há algum tempo. O problema ficou evidente no Memcached 1.4.33 quando o LRU Crawler foi ativado por padrão.

Se você estiver enfrentando esse problema, recomendamos que você desative o LRU Crawler até que haja uma solução. Para fazer isso, use `lru_crawler disable` na linha de comando ou modifique o valor do parâmetro `lru_crawler` (preferencial).

Data de resolução:

Resolução:

## ElastiCache Notificações específicas (RedisOSS)

No momento, não há notificações pendentes ElastiCache (RedisOSS).

# ElastiCache Histórico da documentação

- API versão: 2015-02-02
- 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 desta documentação, você pode assinar o RSS feed.

## ElastiCache Atualizações recentes

Alteração	Descrição	Data
<a href="#">Support for ElastiCache with Valkey</a>	ElastiCache agora suporta Valkey. <a href="#">O Valkey 7.2.6 é compatível com o Redis OSS 7.2</a> Para obter mais informações, consulte <a href="#">Valkey</a> .	8 de outubro de 2024
<a href="#">Tamanho de nós reservados flexíveis</a>	ElastiCache agora suporta <a href="#">nós reservados flexíveis de tamanho</a> . Para obter mais informações, consulte <a href="#">Amazon ElastiCache Pricing</a> .	1 de outubro de 2024
<a href="#">ElastiCache (RedisOSS) adicionou suporte para tamanhos adicionais de nós C7gn</a>	ElastiCache (RedisOSS) adicionou suporte para tamanhos adicionais de nós C7gn.	10 de janeiro de 2024
<a href="#">ElastiCache (RedisOSS) agora suporta a criação de caches sem servidor</a>	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	27 de novembro de 2023

es, consulte [Choosing between deployment options](#).

Como parte desse recurso, [novas permissões](#) foram adicionadas ElastiCacheServiceRolePolicy e para permitir AmazonElastiCacheFullAccess a associação de caches sem servidor com endpoints gerenciados. VPC Além disso, as permissões foram adicionadas para dar suporte a uma experiência revisada do console usando a política AmazonElastiCacheFullAccess .

[ElastiCache \(Memcached\) 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 ElastiCacheServiceRolePolicy e para permitir AmazonElastiCacheFullAccess a associação de caches sem servidor com endpoints gerenciados. VPC 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 \(RedisOSS\) 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 \(RedisOSS\) agora suporta a modificação das configurações de criptografia em trânsito](#)

Agora você pode alterar a TLS configuração dos seus OSS clusters Redis sem a necessidade de reconstruir ou reprovisionar os clusters ou afetar a disponibilidade dos aplicativos. Para obter mais informações, consulte [Habilitação da criptografia em trânsito em um cluster existente](#).

28 de dezembro de 2022

[ElastiCache \(RedisOSS\) agora suporta a autenticação de usuários usando IAM](#)

IAMA autenticação permite que você autentique uma conexão com ElastiCache (RedisOSS) usando AWS IAM identidades. Isso possibilita que você fortaleça seu modelo de segurança e simplifique várias tarefas administrativas de segurança. Para obter mais informações, consulte [Autenticando com IAM](#).

16 de novembro de 2022

[ElastiCache \(RedisOSS\) agora oferece suporte ao Redis 7 OSS](#)

Esta versão traz vários novos recursos para a Amazon ElastiCache (RedisOSS): OSS funções do Redis, ACL melhorias e Sharded Pub/Sub. Para obter mais informações, consulte [ElastiCache \(RedisOSS\) versão 7.0](#).

8 de novembro de 2022

## [ElastiCache \(RedisOSS\) agora suporta IPv6](#)

7 de novembro de 2022

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4eIPv6), 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 OSS mecanismo Redis 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 \(Memcached\) agora suporta IPv6](#)

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4eIPv6), 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](#).

7 de novembro de 2022

## [ElastiCache \(Memcached\) agora oferece suporte à criptografia em trânsito](#)

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 \(TLS\)](#).

26 de maio de 2022



[ElastiCache \(RedisOSS\) agora oferece suporte ao formato nativo JavaScript Object Notation \(JSON\)](#)

O formato nativo JavaScript Object Notation (JSON) é uma forma simples e sem esquemas de codificar conjuntos de dados complexos dentro de clusters do Redis. OSS Você pode armazenar e acessar dados de forma nativa usando o formato JavaScript Object Notation (JSON) dentro dos OSS clusters do Redis e atualizar JSON os dados armazenados nesses clusters, sem precisar gerenciar o código personalizado para serializá-lo e desserializá-lo. Para obter mais informações, consulte [Introdução ao JSON](#).

25 de maio de 2022

[ElastiCache agora suporta PrivateLink](#)

AWS PrivateLink permite que você acesse ElastiCache API as operações de forma privada sem um gateway de internet, NAT dispositivo, VPN conexão ou conexão AWS Direct Connect. Para obter mais informações, consulte [Amazon ElastiCache API e VPC endpoints de interface \(AWS PrivateLink\)](#) para Redis ou OSS [Amazon ElastiCache API e VPC endpoints de interface \(AWS PrivateLink\)](#) para Memcached.

24 de janeiro de 2022

[ElastiCache \(RedisOSS\) agora oferece suporte ao Redis OSS 6.2 e à hierarquização de dados](#)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão do OSS mecanismo Redis suportada pela Amazon. ElastiCache (RedisOSS) 6.2 inclui melhorias de desempenho para clusters TLS habilitados usando tipos de nós x86 com 8 vCPUs ou mais ou tipos de nós Graviton2 com 4 ou mais vCPUs. ElastiCache (RedisOSS) também introduz a classificação por níveis de dados. 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 \(RedisOSS\) versão 6.2 \(aprimorada\)](#) e Armazenamento de [dados](#) em camadas.

23 de novembro de 2021

[Suporte para o Auto Scaling](#)

ElastiCache (RedisOSS) agora oferece suporte ao Auto Scaling. ElastiCache O escalonamento automático (RedisOSS) é a capacidade de aumentar ou diminuir automaticamente os fragmentos ou réplicas desejados em seu serviço ElastiCache (OSSRedis). ElastiCache aproveita o serviço Application Auto Scaling para fornecer essa funcionalidade. Para obter mais informações, consulte Clusters do [Auto Scaling ElastiCache \(Redis OSS\)](#).

19 de agosto de 2021

[Support para entrega de registros do Redis OSS Slow](#)

ElastiCache agora permite que você transmita o Redis OSS SLOWLOG para um dos dois destinos: Amazon Data Firehose ou Amazon CloudWatch Logs. 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 condições que determinam como uma IAM política 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 condições que determinam como uma IAM política 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, assim como faria na nuvem. Para obter mais informações, consulte [Usando Outposts para Redis ou OSS Usando Outposts](#) para Memcached.

8 de outubro de 2020

## [ElastiCache agora suporta Redis 6 OSS](#)

A Amazon ElastiCache (RedisOSS) apresenta a próxima versão do OSS mecanismo Redis suportada 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 \(RedisOSS\) Versão 6.0 \(aprimorada\)](#).

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 \(RedisOSS\) agora oferece suporte à escalabilidade do seu ambiente Redis OSS Cluster para até 500 nós ou 500 fragmentos](#)

O modo Redis OSS Cluster possibilita configurações que você pode usar para particionar seus dados em vários fragmentos e oferece melhor escalabilidade, desempenho e disponibilidade. Esse recurso está disponível na versão 5.0.6 em diante do Amazon ElastiCache (RedisOSS) em todas as AWS regiões e para todos os ambientes de cluster novos e existentes ElastiCache (RedisOSS). Para obter mais informações, consulte [Redis OSS Nodes and Shards](#).

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 \(RedisOSS\) adiciona métricas adicionais da Amazon CloudWatch](#)

ElastiCache (RedisOSS) agora oferece suporte a novas CloudWatch métricas, incluindo PubSubCmds e HyperLogLogBasedCmds. Para ver 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 \(RedisOSS\) agora oferece suporte ao Global Datastore for Redis OSS](#)

O OSS recurso Global Datastore for Redis 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 ElastiCache (RedisOSS) 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 \(RedisOSS\) agora suporta a versão 5.0.6 do Redis OSS](#)

Para obter mais informações, consulte [ElastiCache \(RedisOSS\) Versão 5.0.6 \(aprimorada\)](#).

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 CPU desempenho com a capacidade de aumentar o CPU uso 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 AUTH token em um servidor existente ElastiCache \(RedisOSS\)](#)

ElastiCache (RedisOSS) 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 AWS 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 auto-hospedado na OSS Amazon para a EC2 Amazon. ElastiCache Para obter mais informações, consulte [Migração online para ElastiCache](#).

28 de outubro de 2019

[ElastiCache \(RedisOSS\) introduz o escalonamento vertical on-line para o modo Redis OSS Cluster.](#)

Agora você pode ampliar ou reduzir seu OSS cluster Redis fragmentado sob demanda. ElastiCache (RedisOSS) 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 \(RedisOSS\) agora permite que os usuários usem um único endpoint de leitura para seu cluster Amazon ElastiCache \(RedisOSS\).](#)

Esse recurso permite que você direcione todo o tráfego de leitura para seu cluster ElastiCache (RedisOSS) 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 \(RedisOSS\) 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 Suporte \(RedisOSS\) para até 250 nós por cluster Redis OSS](#)

O limite de nós ou fragmentos pode ser aumentado para um máximo de 250 por cluster ElastiCache (RedisOSS). Para obter mais informações, consulte [Estilhaços](#).

19 de novembro de 2018

[ElastiCache Suporte \(RedisOSS\) para failover automático, backup e restauração em todos os nós T2](#)

ElastiCache (RedisOSS) introduz suporte para failover automático, criação de instantâneos e backup e restauração em todos os nós T2. Para obter mais informações, consulte [ElastiCache \(RedisOSS\) Backup e restauração e instantâneo](#).

19 de novembro de 2018

[ElastiCache Suporte \(RedisOSS\) para nós M5 e R5](#)

ElastiCache (RedisOSS) 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 (RedisOSS) 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 ElastiCache (RedisOSS). Veja também [DecreaseReplicaCount](#) e [IncreaseReplicaCount](#) na ElastiCache API Referência.

17 de setembro de 2018

[Certificação de RAMP conformidade do Fed](#)

ElastiCache (RedisOSS) agora está certificado para RAMP conformidade com o Fed. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

30 de agosto de 2018

[Atualizações do motor Valkey ou Redis OSS \(modo de cluster ativado\)](#)

A Amazon ElastiCache (RedisOSS) 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

[PCIDSScertificação de conformidade](#)

ElastiCache (RedisOSS) agora está certificado quanto à PCI DSS conformidade. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

5 de julho de 2018

[Support for ElastiCache \(RedisOSS\) 4.0.10](#)

ElastiCache (RedisOSS) 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 \(RedisOSS\) Versão 4.0.10 \(aprimorada\)](#).

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 ([ElastiCache \(RedisOSS\) User Guide](#)) e para o Memcached ([ElastiCache \(Memcached\) User Guide](#)). 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 para a ngineCPU Utilization métrica E](#)

ElastiCache (RedisOSS) adicionou uma nova métrica, EngineCPU Utilization, que relata a porcentagem CPU da sua capacidade que está sendo usada atualmente. Para obter mais informações, consulte [Métricas para 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
<p>Suporte para a região Ásia-Pacífico (Osaka-local).</p>	<p>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 obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li> <li>• <a href="#">Tipos de nó de cache compatíveis</a></li> </ul>	<p>12 de fevereiro de 2018</p>
<p>Suporte para a Europa (Paris).</p>	<p>ElastiCache adicionou suporte para a região da UE (Paris). Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li> <li>• <a href="#">Tipos de nó de cache compatíveis</a></li> </ul>	<p>18 de dezembro de 2017</p>



Alteração	Descrição	Alterado em
Suporte para a região da China (Ningxia)	<p>A Amazon ElastiCache adicionou suporte para a região da China (Ningxia). Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li><li>• <a href="#">Tipos de nó de cache compatíveis</a></li></ul>	11 de dezembro de 2017
Suporte para funções vinculadas ao serviço	<p>Esta versão ElastiCache adicionou suporte para funções vinculadas ao serviço (SLR). Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Usando funções vinculadas a serviços para a Amazon ElastiCache</a></li><li>• <a href="#">Configure suas permissões (somente para novos ElastiCache usuários)</a></li></ul>	7 de dezembro de 2017
Suporte para tipos de nós R4	<p>Esta versão ElastiCache adicionou tipos de nós de suporte R4 em todas as AWS regiões suportadas pelo ElastiCache. Você pode adquirir os tipos de nó R4 sob demanda ou como nós de cache reservados. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Tipos de nó de cache compatíveis</a></li><li>• <a href="#">Parâmetros específicos do tipo de nó do Memcached</a></li><li>• <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a></li></ul>	20 de novembro de 2017

Alteração	Descrição	Alterado em
ElastiCache (RedisOSS) 3.2.10 e suporte para refragmentação online	<p>Amazon ElastiCache (RedisOSS) adiciona suporte para ElastiCache (RedisOSS) 3.2.10. ElastiCache (RedisOSS) também introduz o redimensionamento on-line do cluster para adicionar ou remover fragmentos do cluster enquanto ele continua atendendo às solicitações de E/S recebidas. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Redimensionamento de cluster on-line</a></li><li>• <a href="#">Refragmentação online para Valkey ou Redis OSS (modo de cluster ativado)</a></li></ul>	9 de novembro de 2017
HIPAA elegibilidade	<p>ElastiCache (RedisOSS) a versão 3.2.6 agora é certificada para HIPAA elegibilidade quando a criptografia está habilitada em seu cluster. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Validação de conformidade para a Amazon ElastiCache</a></li><li>• <a href="#">Segurança de dados na Amazon ElastiCache</a></li></ul>	2 de novembro de 2017

Alteração	Descrição	Alterado em
ElastiCache (RedisOSS) 3.2.6 e suporte para criptografia	<p>ElastiCache adiciona suporte para ElastiCache (RedisOSS) 3.2.6, que inclui dois recursos de criptografia:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• A criptografia em repouso criptografa dados no disco durante as operações de sincronização e backup.</li> </ul> <p>Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Segurança de dados na Amazon ElastiCache</a></li> <li>• <a href="#">Mecanismos e versões compatíveis</a></li> </ul>	25 de outubro de 2017
Tópico de padrões de conexão	<p>ElastiCache a documentação adiciona um tópico que abrange vários padrões de acesso a um ElastiCache cluster em uma AmazonVPC.</p> <p>Para obter mais informações, consulte <a href="#">Padrões de acesso para acessar um ElastiCache cache em uma Amazon VPC</a> o Guia ElastiCache do usuário.</p>	24 de abril de 2017
Suporte para Memcached 1.4.34	<p>ElastiCache adiciona suporte ao Memcached versão 1.4.34, que incorpora várias correções às versões anteriores do Memcached.</p> <p>Para obter mais informações, consulte as <a href="#">notas de lançamento do Memcached 1.4.34</a> em Memcached on. GitHub</p>	10 de abril de 2017

Alteração	Descrição	Alterado em
Suporte para testar o failover automático	<p>ElastiCache adiciona suporte para testar o failover automático em OSS clusters Redis que oferecem suporte à replicação. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Teste do failover automático</a> no ElastiCache Guia do usuário.</li><li>• <a href="#">TestFailover</a> na ElastiCache APIReferência.</li><li>• <a href="#">test-failover</a> na AWS CLI Referência.</li></ul>	4 de abril de 2017
Restauração aprimorada do Redis OSS	<p>ElastiCache adiciona OSS backup e restauração aprimorados do Redis 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 o API eCLI, esse recurso pode restaurar um número diferente de grupos de nós em vez de um número diferente de fragmentos.) Essa atualização também oferece suporte a diferentes configurações de OSS slot Redis. Para obter mais informações, consulte <a href="#">Restauração de um backup para um novo cache</a>.</p>	15 de março de 2017

Alteração	Descrição	Alterado em
Novo parâmetro de gerenciamento de OSS memória do Redis	<p>ElastiCache adiciona um novo OSS parâmetro <code>Redisreserved-memory-percent</code>, que facilita o gerenciamento da memória reservada. Esse parâmetro está disponível em todas as versões do ElastiCache (RedisOSS). Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Gerenciando a memória reservada para Valkey e Redis OSS</a></li> <li>• <a href="#">Novos parâmetros para o Redis 3.2.4 OSS</a></li> </ul>	15 de março de 2017
Suporte para Memcached 1.4.33	<p>ElastiCache adiciona suporte para a versão 1.4.33 do Memcached. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Memcached versão 1.4.33</a></li> <li>• <a href="#">Parâmetros adicionados do Memcached 1.4.33</a></li> </ul>	20 de dezembro de 2016
Suporte para a região oeste da Europa (Londres)	<p>ElastiCache adiciona suporte para a região da UE (Londres). No momento, há suporte apenas para os tipos de nós T2 e M4. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li> <li>• <a href="#">Tipos de nó de cache compatíveis</a></li> </ul>	13 de dezembro de 2016
Suporte à região Canadá (Montreal)	<p>ElastiCache adiciona suporte para a região do Canadá (Montreal). Somente os tipos de nó M4 e T2 são atualmente suportados nesta AWS região. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li> <li>• <a href="#">Tipos de nó de cache compatíveis</a></li> </ul>	8 de dezembro de 2016

Alteração	Descrição	Alterado em
Suporte para tipos de nós M4 e R3	<p>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 obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li><li>• <a href="#">Tipos de nó de cache compatíveis</a></li></ul>	1º de novembro de 2016
Suporte para a região Leste dos EUA 2 (Ohio)	<p>ElastiCache adiciona suporte para a região Leste dos EUA (Ohio) (us-east-2) com os tipos de nós M4, T2 e R3. Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Supported Regions</a> (Regiões compatíveis)</li><li>• <a href="#">Tipos de nó de cache compatíveis</a></li></ul>	17 de outubro de 2016

Alteração	Descrição	Alterado em
Support para Redis Cluster OSS	<p>ElastiCache adiciona suporte ao Redis OSS Cluster (aprimorado). Os clientes que usam o Redis OSS Cluster 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 Redis OSS Cluster são cerca de um quarto dos das versões anteriores.</p> <p>Este release inclui um console de gerenciamento reprojetoado que usa terminologia de acordo com o uso da indústria.</p> <p>Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"><li>• <a href="#">Comparando Memcached e Redis OSS</a></li><li>• <a href="#">ElastiCache componentes e recursos</a>: observe as seções sobre Nós, Fragmentos, Clusters e Replicação.</li><li>• <a href="#">ElastiCache terminologia</a></li></ul>	12 de outubro de 2016

Alteração	Descrição	Alterado em
Suporte para tipo de nós M4	<p>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 obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Tipos de nó de cache compatíveis</a></li> <li>• <a href="#">Parâmetros específicos do tipo de nó do Memcached</a></li> <li>• <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a></li> </ul>	3 de agosto de 2016
Suporte para a região de Mumbai	<p>ElastiCache adiciona suporte para a região Ásia-Pacífico (Mumbai). Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Tipos de nó de cache compatíveis</a></li> <li>• <a href="#">Parâmetros específicos do tipo de nó do Memcached</a></li> <li>• <a href="#">Parâmetros específicos do tipo de OSS nó Redis</a></li> </ul>	27 de junho de 2016
Exportação de snapshots	<p>ElastiCache adiciona a capacidade de exportar um OSS snapshot do Redis para que você possa acessá-lo de fora. ElastiCache Para obter mais informações, consulte as informações a seguir.</p> <ul style="list-style-type: none"> <li>• <a href="#">Exportação de um backup</a>no Guia do ElastiCache usuário da Amazon</li> <li>• <a href="#">CopySnapshot</a>na Amazon ElastiCache API Reference</li> </ul>	26 de maio de 2016



Alteração	Descrição	Alterado em
Ampliação do tipo de nó	ElastiCache adiciona a capacidade de escalar seu tipo de OSS nó Redis. Para obter mais informações, consulte <a href="#">Dimensionamento ElastiCache</a> .	24 de março de 2016
Atualização fácil do mecanismo	ElastiCache adiciona a capacidade de atualizar facilmente seu mecanismo de OSS cache Redis. Para obter mais informações, consulte <a href="#">Gerenciamento de versões para ElastiCache</a> .	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 <a href="#">Supported cache node types</a> .	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 <a href="#">Outros ElastiCache tutoriais e vídeos</a> .	12 de fevereiro de 2016
Support para Redis 2.8.24 OSS	ElastiCache adiciona suporte para a OSS versão 2.8.24 do Redis com melhorias adicionadas desde o Redis 2.8.23. OSS Entre os aprimoramentos incluem correções de bugs e suporte para o registro em log de endereços de acesso à memória incorretos. Para obter mais informações, consulte as informações a seguir. <ul style="list-style-type: none"> <li>• <a href="#">ElastiCache (RedisOSS) versão 2.8.24 (aprimorada)</a></li> <li>• <a href="#">Notas de lançamento do Redis OSS 2.8</a></li> </ul>	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 OSS versões mais recentes do Redis oferecem uma experiência de usuário melhor e mais estável, OSS as versões 2.6.13, 2.8.6 e 2.8.19 do Redis não estão mais listadas no Management Console. ElastiCache Para outras opções e mais informações, consulte <a href="#">Mecanismos e versões compatíveis</a> .	15 de dezembro de 2015
Support para Redis OSS 2.8.23.	ElastiCache adiciona suporte para a OSS versão 2.8.23 do Redis com melhorias adicionadas desde o Redis 2.8.22. OSS 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 <a href="#">ElastiCache (RedisOSS) versão 2.8.23 (aprimorada)</a> .	13 de novembro de 2015

Alteração	Descrição	Alterado em
Support para Redis OSS 2.8.22.	<p>ElastiCache adiciona suporte para a OSS versão 2.8.22 do Redis com aprimoramentos e melhorias ElastiCache adicionais desde a versão 2.8.21. Entre os aprimoramentos estão:</p> <ul style="list-style-type: none"><li data-bbox="402 457 1117 709">• 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.</li><li data-bbox="402 751 1117 846">• CloudWatch Métricas adicionais — SaveInProgressReplicationBytes.</li><li data-bbox="402 888 1117 1024">• Para habilitar sincronizações parciais, o OSS parâmetro Redis <code>repl-backlog-size</code> agora se aplica a todos os clusters.</li></ul> <p>Para obter uma lista completa de mudanças e mais informações, consulte <a href="#">ElastiCache (RedisOSS) versão 2.8.22 (aprimorada)</a>.</p> <p>Esta versão da documentação inclui uma reorganização da documentação e a remoção da documentação da interface de linha de ElastiCache comando (CLI). Para usar a linha de comando, consulte a <a href="#">Linha de AWS comando</a> para ElastiCache.</p>	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 o gerenciamento de cache (LRU) usado menos recentemente como uma tarefa em segundo plano, a escolha de jenkins ou murmur3 como seu algoritmo de hash, novos comandos e diversas correções de bugs. Para obter mais informações, as <a href="#">Notas de release do Memcached</a>	27 de agosto de 2015
Support para Memcached Auto Discovery usando 5.6 PHP	Esta versão da Amazon ElastiCache adiciona suporte ao cliente Memcached Auto Discovery para a PHP versão 5.6. Para obter mais informações, consulte <a href="#">Compilando o código-fonte para o cliente de ElastiCache cluster para PHP</a> .	29 de julho de 2015
Support para Redis 2.8.21 OSS	ElastiCache adiciona suporte para a OSS versão 2.8.21 do Redis e OSS melhorias do Redis desde a versão 2.8.19. Esta OSS versão do Redis inclui várias correções de bugs. Para obter mais informações, consulte as <a href="#">notas de versão do Redis OSS 2.8</a> .	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 <a href="#">Acessando ElastiCache de fora AWS</a> .	9 de julho de 2015

Alteração	Descrição	Alterado em
Mensagens de substituição de nó adicionadas	<p>ElastiCache adiciona três mensagens relacionadas à substituição programada do nó, ElastiCache: NodeReplacementScheduled, ElastiCache: NodeReplacementRescheduled, ElastiCache: e ElastiCache: NodeReplacementCanceled.</p> <p>Para obter mais informações e ações que você pode tomar quando um nó está programado para ser substituído, consulte ElastiCache <a href="#">Notificações de eventos e Amazon SNS</a>.</p>	11 de junho de 2015

Alteração	Descrição	Alterado em
Support for Redis OSS v. 2.8.19.	<p>ElastiCache adiciona suporte para a OSS versão 2.8.19 do Redis e OSS melhorias do Redis desde a versão 2.8.6. Este suporte inclui suporte para:</p> <ul style="list-style-type: none"><li>• A estrutura de HyperLogLog dados, com os OSS comandos Redis PFADDPFCOUNT, e. PFMERGE</li><li>• Consultas de alcance lexicográfico com os novos comandos ZRANGEBYLEX, e. ZLEXCOUNT ZREMRANGEBYLEX</li><li>• Introduziu várias correções de erros, ou seja, impedir que um nó primário envie dados obsoletos para nós de réplica ao falhar no primário SYNC quando um processo secundário de salvamento em segundo plano (bgsave) é encerrado inesperadamente.</li></ul> <p>Para obter mais informações sobre HyperLogLog, consulte a <a href="#">OSS nova estrutura de dados do Redis: a. HyperLogLog</a></p> <p>Para obter mais informações sobre PFADD, PFCOUNT, e PFMERGE, consulte a <a href="#">OSS documentação do Redis</a> e clique em HyperLogLog.</p>	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 <a href="#">Monitoramento de custos com tags de alocação de custos</a> .	9 de fevereiro de 2015

Alteração	Descrição	Alterado em
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
Suporte para a região Europa (Frankfurt)	ElastiCache adiciona suporte para a região Europa (Frankfurt) (eu-central-1).	19 de janeiro de 2015
Suporte 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 RedisOSS. 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 <a href="#">Minimizando o tempo de inatividade ElastiCache usando o Multi-AZ com Valkey e Redis OSS</a> .	24 de outubro de 2014
AWS CloudTrail I registro de API chamadas suportado	ElastiCache adiciona suporte para uso AWS CloudTrail para registrar todas as ElastiCache API chamadas. Para obter mais informações, consulte <a href="#">Registrando ElastiCache API chamadas da Amazon com AWS CloudTrail</a> .	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 <a href="#">Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros</a> .	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 <a href="#">Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros</a> .	1 de julho de 2014
PHPdescoberta automática	Foi adicionado suporte para a descoberta automática da PHP versão 5.5.	13 de maio de 2014
Backup e restauração para clusters Redis OSS	Nesta versão, ElastiCache permite que os clientes criem instantâneos de seus OSS clusters do Redis 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 em todos os dados no cache do RedisOSS. 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 <a href="#">Snapshots e restauração</a> .	24 de abril de 2014
Redis 2.8.6 OSS	ElastiCache suporta o Redis OSS 2.8.6, além do Redis 2.6.13. OSS Com o Redis OSS 2.8.6, os clientes podem melhorar a resiliência e a tolerância a falhas das réplicas de leitura, com suporte para resincronização parcial e um número mínimo de réplicas de leitura definido pelo usuário que devem estar sempre disponíveis. 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 OSS cache Redis	<p>ElastiCache oferece o software de mecanismo de OSS cache Redis, além do Memcached. Os clientes que atualmente usam o Redis OSS podem “semear” um novo cluster de OSS cache do ElastiCache Redis com seus dados existentes a partir de um arquivo de OSS instantâneo do Redis, facilitando a migração para um ambiente gerenciado. ElastiCache</p> <p>Para oferecer suporte aos recursos de OSS replicação do Redis, o ElastiCache API agora oferece suporte a grupos de replicação. Os clientes podem criar um grupo de replicação com um nó de OSS cache primário do Redis e adicionar um ou mais nós de réplica de leitura que permanecem automaticamente sincronizados com os 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.</p>	3 de setembro de 2013
Support para Amazon Virtual Private Cloud padrão (VPC)	<p>Nesta versão, ElastiCache está totalmente integrado com a Amazon Virtual Private Cloud (VPC). Para novos clientes, os clusters de cache são criados em uma Amazon VPC por padrão. Para obter mais informações, consulte <a href="#">Amazon VPCs e ElastiCache segurança</a>.</p>	8 de janeiro de 2013
PHPsuporte para descoberta automática de nós de cache	<p>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 o.</p>	2 de janeiro de 2013

Alteração	Descrição	Alterado em
Support para Amazon Virtual Private Cloud (VPC)	Nesta versão, os ElastiCache clusters podem ser lançados na Amazon Virtual Private Cloud (VPC). Por padrão, os clusters de cache de novos clientes são criados VPC automaticamente em uma Amazon; os clientes existentes podem migrar para a Amazon VPC em seu próprio ritmo. Para obter mais informações, consulte <a href="#">Amazon VPCs e ElastiCache segurança</a> .	20 de dezembro de 2012
Descoberta automática de nós de cache e nova versão do mecanismo de cache	<p>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.</p> <p>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 <a href="#">Configurando os parâmetros do motor usando grupos de ElastiCache parâmetros</a>.</p>	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.