



Guia do usuário

Amazon ElastiCache para Redis



Versão da API 2015-02-02

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

Amazon ElastiCache para Redis: 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 desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é ElastiCache para o Redis?	1
Armazenamento em cache sem servidor	1
Clusters autoprojetoados	2
Serviços relacionados	2
Como funciona	3
Cache e mecanismos de cache	3
Escolha da implantação	8
Comparando recursos	10
Recursos do ElastiCache	15
AWSRegiões e zonas de disponibilidade da	17
Casos de uso	18
Armazenamento de dados na memória	19
Placares de jogos (conjuntos classificados do Redis)	20
Mensagens (publicação/assinatura do Redis)	22
Dados de recomendação (hashes do Redis)	25
Outros usos do Redis	25
Depoimentos de clientes do ElastiCache	25
Introdução ao ElastiCache for Redis	26
Configuração	26
Inscreva-se para um Conta da AWS	26
Criar um usuário administrativo	27
Conceder acesso programático	28
Configurar permissões	30
Configurar o EC2	31
Conceder acesso de rede	31
Configurar o redis-cli	32
Criar um cache	33
Ler e gravar dados	34
Limpeza	36
Próximos Passos	37
Conceitos básicos do ElastiCache e dos SDKs da AWS	37
Python e ElastiCache	37
Tutorial: Configurando uma função Lambda para acessar a Amazon em uma ElastiCache Amazon VPC	55

Etapa 1: criar um cache sem servidor	55
Etapa 2: criar uma função do Lambda	58
Etapa 3: testar a função do Lambda	62
Etapa 4: limpar (opcional)	63
Projeto do próprio cluster do ElastiCache	65
Componentes e recursos	65
Nodes	66
ElastiCache para fragmentos do Redis	66
ElastiCache para clusters Redis	67
ElastiCache para replicação do Redis	69
AWS Regiões e zonas de disponibilidade	71
ElastiCache para endpoints Redis	72
Grupos de parâmetros	73
ElastiCache para segurança do Redis	73
Grupos de sub-redes	73
ElastiCache para backups do Redis	74
Eventos	74
Terminologia do ElastiCache for Redis	75
Projeto do próprio cluster	78
Configuração	78
Etapa 1: criar um grupo de sub-redes	78
Etapa 2: criar um cluster	81
Etapa 3: autorizar o acesso ao cluster	89
Etapa 4: conexão ao nó do cluster	92
Etapa 5: excluir um cluster	100
Tutoriais e vídeos	101
O que faço agora?	107
Gerenciamento de nós	108
Visualizando o status ElastiCache do nó	109
Nós e fragmentos do Redis	114
Conexão a nós	117
Tipos de nó compatíveis	120
Reinicialização de nós (somente para o modo cluster desabilitado)	131
Substituição de nós	133
Nós reservados	140
Migração de nós da geração anterior	152

Gerenciamento de clusters	155
Escolher um tipo de rede	157
Classificação de dados em níveis	161
Preparação de um cluster	168
Criar um cluster	176
Visualização dos detalhes de um cluster	187
Modificar um cluster	200
Adição de nós a um cluster	206
Remoção de nós de um cluster	214
Cancelamento de operações de adição ou exclusão de nós pendentes	222
Excluir um cluster	223
Acesso ao seu cluster ou grupo de replicação	226
Localização de endpoints de conexão	232
Fragmentos	244
Comparação dos caches autoprojeto do Memcached e do Redis	249
Migração online para o ElastiCache	254
Visão geral	255
Etapas da migração	255
Preparação dos nós do Redis de origem e de destino para a migração	256
Testar a migração de dados	257
Início da migração	258
Verificação do andamento do processo de migração de dados	260
Conclusão da migração de dados	260
Execução de migração de dados online usando o console	261
Escolher regiões e zonas de disponibilidade	262
Localização dos seus nós	264
Regiões e endpoints com suporte	264
Uso de zonas locais	269
Uso do Outposts	271
Trabalhando com ElastiCache	275
Snapshots e restauração	275
Restrições	276
Impacto sobre o desempenho dos backups de clusters autoprojeto	277
Programação de backups automáticos	278
Realização de backups manuais	279
Criação de um backup final	285

Descrição de backups	288
Copiar backups	290
Exportação de um backup	292
Restauração de um backup	300
Exclusão de um backup	302
Marcação de backups	303
Propagação de um novo cluster autoprotetado com um backup	305
Versões de mecanismos e atualização	315
Versões de mecanismos e atualização	316
Versões do Redis com suporte	321
Cronograma de fim de vida das versões do Redis	335
Como atualizar as versões dos mecanismos	319
Resolução de atualizações de mecanismo bloqueadas	319
Principais diferenças de comportamento e compatibilidade da versão	338
Práticas recomendadas e estratégias de armazenamento em cache	342
Trabalhar com o Redis	343
Práticas recomendadas com clientes do Redis	383
Práticas recomendadas durante o trabalho com clusters autoprotetados	410
Práticas recomendadas do Redis	423
Estratégias de armazenamento em cache	425
Gerenciamento do cluster autoprotetado	430
Auto Scaling ElastiCache para clusters Redis	431
Modificar o modo de cluster	479
Replicação em regiões da AWS usando o armazenamentos de dados globais	482
Alta disponibilidade com o uso de grupos de replicação	510
Gerenciamento da manutenção	598
Configuração de parâmetros do mecanismo usando grupos de parâmetros	601
Dimensionamento ElastiCache para Redis	700
Dimensionamento sem servidor ElastiCache	700
Definição dos limites de escalabilidade para gerenciar custos	701
Pré-escalamento com Serverless ElastiCache	701
Definindo limites de escala usando o console e AWS CLI	702
Dimensionamento ElastiCache para clusters do Redis projetados pelo próprio usuário	704
Conceitos básicos do JSON no ElastiCache para Redis	775
Visão geral do tipo de dados do JSON do Redis	776
Comandos JSON	788

Marcação de recursos do ElastiCache	829
Monitoramento de custos com tags	841
Gerenciar tags usando o AWS CLI	843
Gerenciamento de tags usando a API do ElastiCache	846
Lente do Well-Architected para o Amazon ElastiCache	849
Pilar Excelência operacional	850
Pilar Segurança	858
Pilar Confiabilidade	865
Pilar Eficiência de performance	871
Pilar Otimização de custos	882
Solução de problemas	889
Problemas de conectividade	889
Erros do cliente Redis	890
Solução de problemas de alta latência no Serverless ElastiCache	890
Solução de problemas de limitação no Serverless ElastiCache	892
Related Topics	893
Etapas adicionais de solução de	893
Grupos de segurança	893
Network ACLs	894
Tabelas de rotas	896
Resolução do DNS	896
Identificação de problemas com o diagnóstico do lado do servidor	896
Validação da conectividade de rede	902
Limites relacionados à rede	905
Uso da CPU	906
Conexões sendo encerradas do lado do servidor	909
Solução de problemas no lado do cliente para instâncias do Amazon EC2	911
Dissecação do tempo necessário para concluir uma única solicitação	912
Segurança	915
Proteção de dados	916
Segurança de dados no Amazon ElastiCache	916
Privacidade do tráfego entre redes	989
Segurança do Amazon VPCs e do ElastiCache	989
API do Amazon ElastiCache e endpoints da VPC de interface (AWS PrivateLink)	1014
Sub-redes e grupos de sub-redes	1017
Identity and Access Management	1026

Público	1026
Autenticando com identidades	1027
Como gerenciar acesso usando políticas	1031
Como o Amazon ElastiCache funciona com o IAM	1033
Exemplos de políticas baseadas em identidade	1041
Solução de problemas	1044
Controle de acesso	1046
Visão geral do gerenciamento de acesso	1047
Validação de compatibilidade	1090
Mais informações	1092
Resiliência	1092
Atenuar falhas	1092
Segurança da infraestrutura	1096
Atualizações de serviço	1096
Gerenciando atualizações de serviços	1097
Vulnerabilidades de segurança abordadas	1102
Registrar em log e monitoramento	1105
Métricas e eventos sem servidor	1105
Métricas sem servidor	1105
Eventos sem servidor	1115
Métricas e eventos de clusters autoprojetoados	1128
Métricas de cluster autoprojetoado	1129
Eventos de cluster autoprojetoados	1129
Entrega de logs	1138
Monitoramento do uso	1151
Monitoramento de eventos do Amazon SNS	1181
Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail	1198
Informações sobre o Amazon ElastiCache no CloudTrail	1198
Noções básicas sobre entradas de arquivo de log do Amazon ElastiCache	1199
Cotas	1204
Referência	1206
Uso da API do ElastiCache	1206
Como usar a API de consulta	1206
Bibliotecas disponíveis	1210
Solução de problemas de aplicações	1210
Configurar a AWS CLI para o ElastiCache	1211

Pré-requisitos	1212
Obtenção das ferramentas da linha de comando	1214
Configuração das ferramentas	1214
Como fornecer credenciais para as ferramentas	1215
Variáveis de ambiente	1216
Mensagens de erro	1217
Notificações	1219
Notificações gerais do ElastiCache	1219
Notificações específicas do ElastiCache for Redis	1219
ElastiCache para o histórico da documentação do Redis	1220
AWS Glossário	1256
.....	mccclvii

O que é o Amazon ElastiCache for Redis?

Bem-vindo ao Guia do usuário do Amazon ElastiCache for Redis. 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 Redis e Memcached. Use o guia para o mecanismo pelo qual está interessado. Se não souber qual mecanismo deseja usar, consulte [Comparação dos caches autoprojeto do Memcached e do Redis](#) neste guia.

Armazenamento em cache sem servidor

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

ElastiCache para Redis Serverless também elimina a necessidade de planejar e gerenciar a capacidade de armazenamento em cache. ElastiCache for Redis 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 for Redis oferece uma experiência de endpoint simples para desenvolvedores, abstraindo a infraestrutura de cache subjacente e o design do cluster. ElastiCache for Redis 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 for Redis Serverless é compatível com Redis 7.1 e superior.

Projetando seu próprio ElastiCache cluster para Redis

Se você precisar de um controle refinado sobre seu cluster ElastiCache for Redis, poderá optar por criar seu próprio cluster Redis com ElastiCache. 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 de 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 cluster ElastiCache para Redis 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 Redis em modo de cluster habilitando a escala 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 cluster do Redis.

Ao criar seu próprio cluster ElastiCache para Redis, você pode optar por executar o Redis 3.0 ou superior.

Serviços relacionados

[Amazon MemoryDB para Redis](#)

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

- O ElastiCache para Redis é um serviço comumente usado para armazenar dados de outros bancos de dados e armazenamentos de dados usando o Redis. Você deve considerar o ElastiCache para Redis para armazenar workloads em cache nas quais você deseja acelerar o acesso aos dados com seu banco de dados principal ou armazenamento de dados existente (desempenho de leitura e gravação em microssegundos). Você também deve considerar o ElastiCache para Redis para casos de uso em que você deseja usar as estruturas de dados e APIs do Redis para acessar dados armazenados em um banco de dados ou armazenamento de dados primário.

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

[Amazon RDS](#)

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

Como funciona

Aqui você pode encontrar uma visão geral dos principais componentes de uma ElastiCache implantação do Redis.

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 for Redis oferece duas opções de implantação: clusters sem servidor e clusters autoprojeto. Consulte [Escolha entre opções de implantação](#)

Note

A Amazon ElastiCache trabalha com os mecanismos Redis e Memcached. Use o guia para o mecanismo pelo qual está interessado. Se não souber qual mecanismo deseja usar, consulte [Comparação dos caches autoprojeto do Memcached e do Redis](#) neste guia.

Tópicos

- [Como funciona ElastiCache o Redis](#)

- [Dimensões de preço](#)
- [ElastiCache para backups do Redis](#)

Como funciona ElastiCache o Redis

ElastiCache para Redis Serverless

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

Note

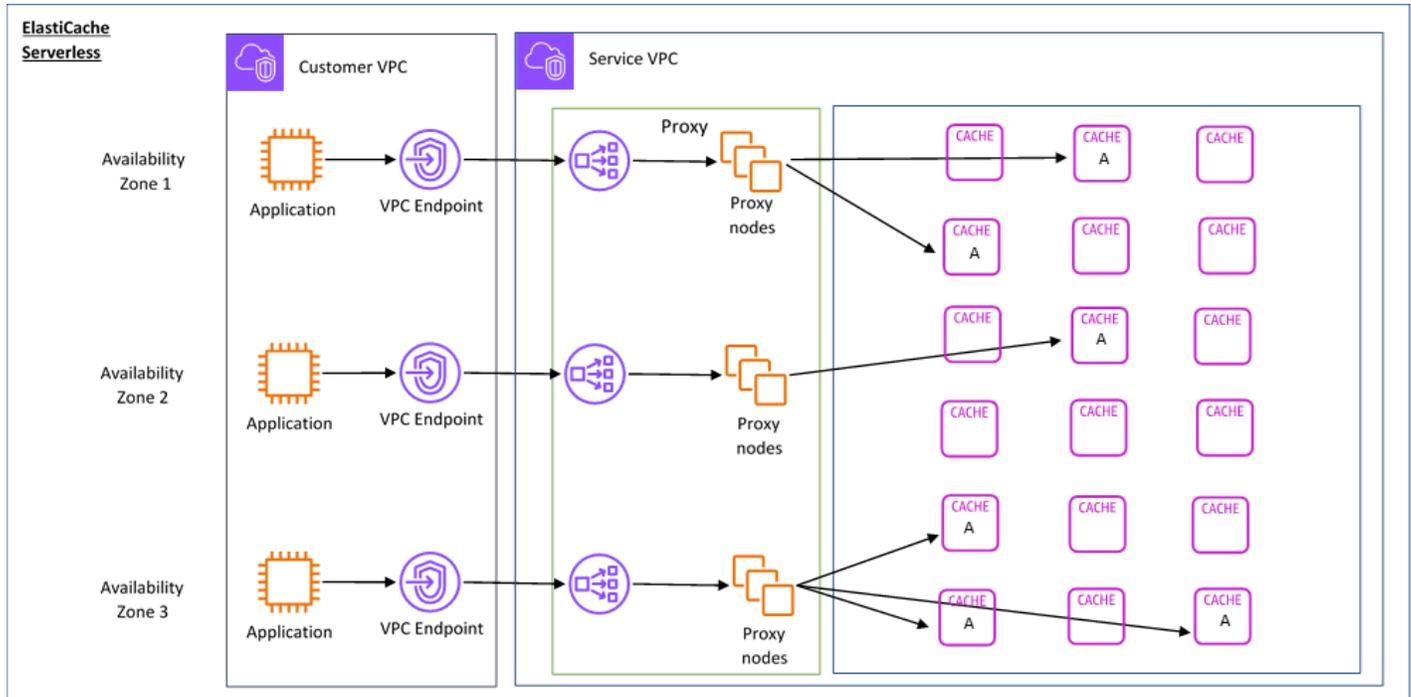
ElastiCache for Redis Serverless executa o Redis no modo de cluster e só é compatível com clientes Redis que oferecem suporte ao TLS e ao protocolo de cluster Redis.

Benefícios principais

- **Sem planejamento de capacidade:** o ElastiCache Serverless elimina a necessidade de planejar a capacidade. ElastiCache O Serverless monitora continuamente a utilização de memória, computação e largura de banda de rede do seu cache e escala vertical e horizontalmente. Ele permite que um nó de cache cresça e, paralelamente, inicia uma operação de aumento de escala horizontal para garantir que o cache possa escalar para atender aos requisitos da aplicação em todos os momentos.
- **Pay-per-use:** Com o ElastiCache Serverless, você paga pelos dados armazenados e pela computação utilizada pela sua carga de trabalho no cache. Consulte [Dimensões de preço](#).
- **Alta disponibilidade:** o ElastiCache Serverless replica automaticamente seus dados em várias zonas de disponibilidade (AZ) para obter alta disponibilidade. Ele monitora automaticamente os nós de cache subjacentes e os substitui em caso de falhas. Ele oferece um SLA de disponibilidade de 99,99% para cada cache.
- **Atualizações automáticas de software:** o ElastiCache Serverless atualiza automaticamente seu cache para a última versão secundária e corrige o software sem nenhum impacto na disponibilidade do seu aplicativo. Quando uma nova versão principal do Redis estiver disponível, você ElastiCache receberá uma notificação.

- **Segurança:** o ElastiCache Sem Servidor sempre criptografa os dados em trânsito e em repouso. É possível usar uma chave gerenciada pelo serviço ou usar a própria chave gerenciada pelo cliente para criptografar dados em repouso.

O diagrama a seguir ilustra como o ElastiCache Serverless funciona.



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

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

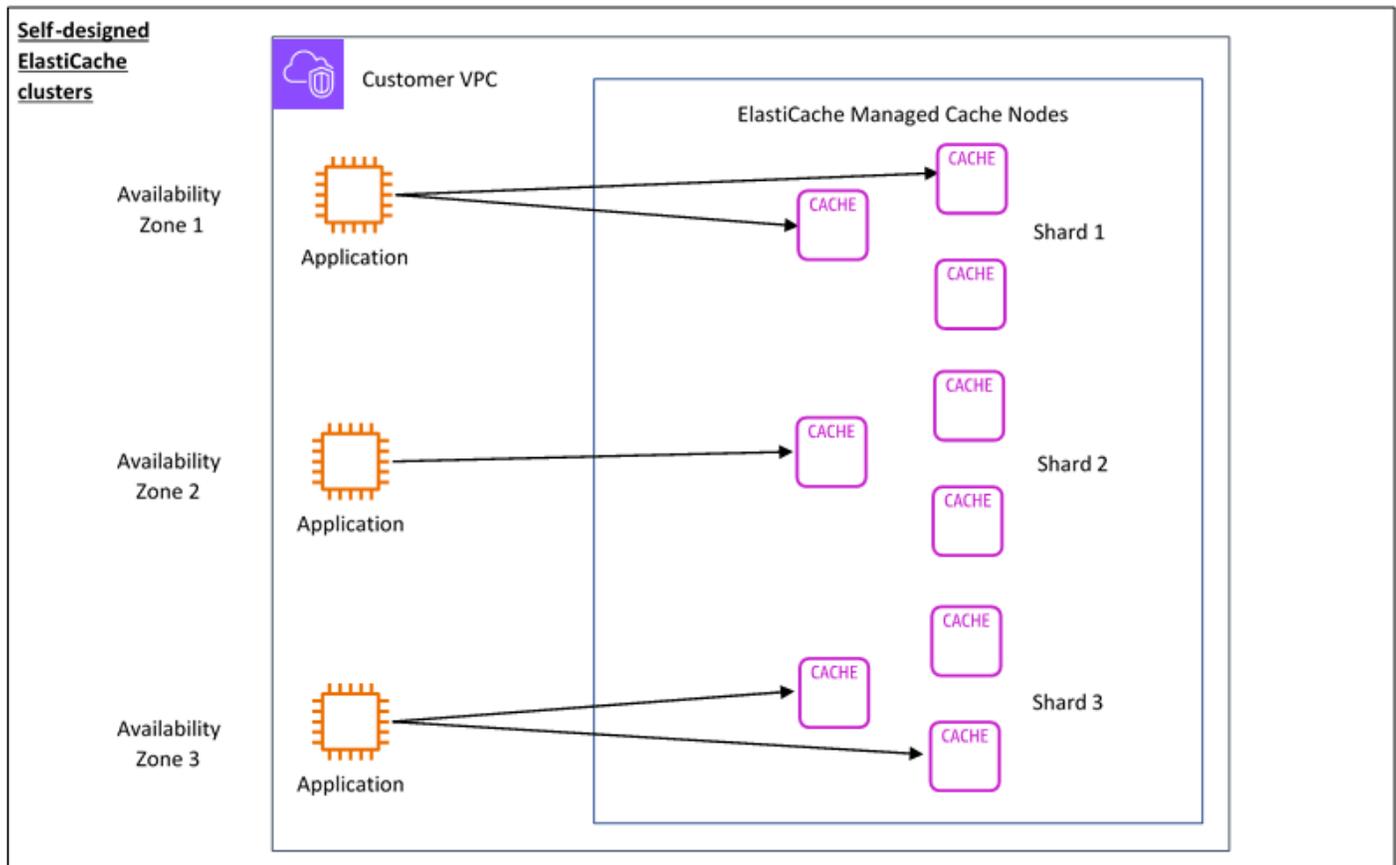
Clusters autoprojetados ElastiCache

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 Redis em modo de cluster criando um cluster com vários fragmentos ou em 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, é possível projetar o cluster com nós em várias 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, é possível usar [Parâmetros específicos do Redis](#) para configurar o mecanismo de cache.
- **Escalar vertical e horizontalmente:** você pode optar por escalar manualmente o cluster aumentando ou diminuindo o tamanho do nó de cache quando necessário. Você também pode escalar horizontalmente ao adicionar novos fragmentos ou adicionar mais réplicas aos fragmentos. 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 uso de CPU e memória no cache.

O diagrama a seguir ilustra como os clusters ElastiCache autoprojetados funcionam.



Dimensões de preço

Você pode implantar ElastiCache em duas opções de implantação. Ao implantar o ElastiCache Serverless, você paga pelo uso dos dados armazenados em GB/hora e pela computação em ElastiCache unidades de processamento (ECPUs). Ao escolher criar seus próprios clusters ElastiCache para Redis, 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 do Redis que seu aplicativo executa em Unidades de ElastiCache Processamento ElastiCache Sem Servidor (eCPUs), uma unidade que inclui o tempo de vCPU e os dados transferidos.

- Leituras e gravações simples exigem um ECPU para cada quiloobyte (KB) de dados transferidos. Por exemplo, um comando GET que transfere até 1 KB de dados consome uma ECPU. Uma solicitação SET que transfere 3,2 KB de dados vai consumir 3,2 eCPUs.
- Comandos que exigem tempo de vCPU adicional consumirão proporcionalmente mais eCPUs. Por exemplo, se usar o [comando HMGET](#) do Redis e consumir três vezes mais tempo de vCPU do que um simples comando SET/GET, a aplicação consumirá três eCPUs.
- Comandos que consomem mais tempo de vCPU e transferem mais dados consomem eCPUs com base na maior das duas dimensões. Por exemplo, se usar o comando HMGET e consumir três vezes mais tempo de vCPU do que um simples comando SET/GET e transferir 3,2 KB de dados, a aplicação consumirá 3,2 eCPUs. Como alternativa, se só transferir 2 KB de dados, ela consumirá três eCPUs.

ElastiCache O Serverless emite uma nova métrica chamada `ElastiCacheProcessingUnits` que ajuda você a entender as eCPUs consumidas pela sua carga de trabalho.

Horas de nó

Você pode optar por projetar o próprio cluster de cache do Redis escolhendo a família de nós do EC2, 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 para backups do Redis

Um backup é uma point-in-time cópia de um cache Redis. 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 autoprojeto

Para obter uma lista dos comandos compatíveis com ambos, consulte [Comandos do Redis compatíveis e restritos](#).

Armazenamento em cache sem servidor

O Amazon ElastiCache Serverless simplifica a criação de cache e escala instantaneamente para dar suporte aos aplicativos mais exigentes dos clientes. Com o ElastiCache Serverless, você pode criar um cache altamente disponível e escalável em menos de um minuto, eliminando a necessidade de provisionar, planejar e gerenciar a capacidade do cluster 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 (SLA) de 99,99% de disponibilidade. Os backups são compatíveis entre si e podem ser exportados e restaurados a partir de clusters autoprojeto.

Clusters autoprojeto

Se você precisar de um controle refinado sobre seu cluster ElastiCache for Redis, poderá optar por criar seu próprio cluster Redis 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 autoprojeto podem ser projetados para fornecer um SLA de disponibilidade de até 99,99%. Os backups são compatíveis entre si e podem ser exportados e restaurados a partir de caches Serverless.

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 deseja um controle mais refinado sobre o tipo de nó que executa o Redis, o número de nós e o posicionamento dos 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 autoprojetados

Atributo	Armazenamento em cache sem servidor	Clusters autoprojetados
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
Compatível com ElastiCache a versão Redis	ElastiCache para Redis versão 7.1 e superior	ElastiCache para Redis versão 4.0 e superior
Modo de cluster	Opera <code>cluster mode enabled</code> somente o Redis em. Os clientes Redis devem oferecer suporte <code>cluster mode enabled</code> para se conectar ao ElastiCache Serverless.	Pode ser configurado para operar no modo cluster ativado ou no modo cluster desativado.
Escalabilidade	Dimensiona automaticamente vertical e horizontalmente sem nenhum gerenciamento de capacidade.	Fornecer controle sobre o dimensionamento, além de exigir monitoramento para garantir que a capacidade atual atenda adequadamente à demanda.

Atributo	Armazenamento em cache sem servidor	Clusters autoprojutados
		<p>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.</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 uso de CPU e memória no cache.</p>
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.

Atributo	Armazenamento em cache sem servidor	Clusters autoprojutados
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 autoprojutados oferecem opções de configuração refinadas. Os clientes podem usar grupos de parâmetros para um controle refinado. Para uma tabela desses valores de parâmetro por tipo de nó, consulte Parâmetros específicos de node-type do Redis .
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). Para clusters Multi-AZ, os dados são replicados de forma assíncrona em várias zonas de disponibilidade para maior disponibilidade e maior latência de leitura.
Criptografia inativa	Sempre ativado. Os clientes podem usar uma chave 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 Chave gerenciada pela AWS ou uma chave gerenciada pelo cliente AWS KMS.
Criptografia em trânsito (TLS)	Sempre ativado. Os clientes devem oferecer suporte à conectividade TLS.	Opção para ativar ou desativar .

Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
Backups	<p>Suporta backups automáticos e manuais de caches sem impacto no desempenho.</p> <p>Os backups são compatíveis entre si e podem ser restaurados em um cache ElastiCache sem servidor ou em um cluster autoprotetido.</p>	<p>Suporta backups automáticos e manuais. Os clusters podem ter algum impacto no desempenho, dependendo da memória reservada disponível. Para ter mais informações, consulte Gerenciamento da memória reservada.</p> <p>Os backups são compatíveis entre si e podem ser restaurados em um cache ElastiCache sem servidor ou em um cluster autoprotetido.</p>
Monitoramento	<p>Support métricas em nível de cache, incluindo taxa de acerto do cache, taxa de erro do cache, tamanho dos dados e eCPUs consumidas.</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 ter mais informações, consulte Eventos de cache sem servidor.</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 notificações de SNS para eventos significativos. Consulte Métricas para Redis.</p>

Atributo	Armazenamento em cache sem servidor	Clusters autoprotetidos
Disponibilidade	Acordo de Nível de Serviço (SLA) de 99,99% de disponibilidade	Clusters autoprotetidos podem ser projetados para atingir um Acordo de Nível de Serviço (SLA) de disponibilidade de até 99,99%, dependendo da configuração.
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 o definidos pelo cliente. Os clientes também podem optar por aplicar um upgrade de versão secundária ou de patch sob demanda.
Armazenamento de dados global	Não suportado	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	Não suportado	Os clusters projetados usando nós da família r6gd têm seus dados em camadas entre a memória e o armazenamento SSD (unidades de estado sólido) local. O armazenamento de dados em camadas fornece uma opção de preço-desempenho para cargas de trabalho do 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 de ElastiCache Processamento (ECPU). Consulte detalhes de preço aqui .	Pay-per-hour, com base no uso do nó de cache. Consulte detalhes de preço aqui .

Tópicos relacionados:

- [Projeto e gerenciamento do próprio cluster do ElastiCache](#)

Recursos do Amazon ElastiCache

Recomendamos que você comece lendo as seguintes seções e consulte-as conforme necessário:

- Destaques do serviço e definição de preço: a [página de detalhes do produto](#) fornece uma visão geral do ElastiCache, destaques de serviços e preços.

- Vídeos do ElastiCache: a seção [Vídeos do ElastiCache](#) tem vídeos que apresentam o Amazon ElastiCache. Os vídeos abordam casos de uso comuns do ElastiCache e demonstram como usar o ElastiCache para reduzir a latência e melhorar o throughput de suas aplicações.
- Conceitos básicos a seção [Começando a usar o Amazon ElastiCache for Redis](#) 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.
- Performance em escala o whitepaper [Performance at Scale with Amazon ElastiCache](#) discute estratégias de armazenamento em cache que ajudam sua aplicação a apresentar boa performance em grande escala.

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

- [Documentação da AWS Command Line Interface](#)

Esta seção fornece informações sobre como baixar a AWS CLI, fazer a AWS CLI funcionar no seu sistema e fornecer suas credenciais da AWS.

- [Documentação da AWS CLI do ElastiCache](#)

Este documento separado abrange tudo da AWS CLI para comandos do ElastiCache, incluindo sintaxe e exemplos.

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

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

A Amazon Web Services fornece uma série de kits de desenvolvimento de software (SDKs) com suporte para o ElastiCache. Você pode codificar para o ElastiCache usando Java, .NET, PHP, Ruby e outras linguagens. Esses SDKs podem simplificar muito o desenvolvimento da sua aplicação, formatando suas solicitações para o ElastiCache, analisando respostas e fornecendo a lógica de repetição e o tratamento de erros.

- [Uso da API do ElastiCache](#)

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

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

Este documento à parte abrange todas as operações da API do ElastiCache, incluindo a sintaxe e exemplos.

AWSRegiões e zonas de disponibilidade da

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

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

Você pode criar seu cluster em várias zonas de disponibilidade, uma opção chamada de implantação multi-AZ. Quando você escolhe essa opção, a Amazon automaticamente provisiona e mantém uma instância de nó secundário em espera em uma zona de disponibilidade diferente. A instância do seu nó primário é sincronicamente replicada ao longo das zonas de disponibilidade para a instância secundária. Essa abordagem ajuda a fornecer redundância de dados e suporte a failover, eliminar congelamentos de E/S e minimizar picos de latência durante backups do sistema. Para obter mais informações, consulte [Minimização do tempo de inatividade no ElastiCache for Redis com o multi-AZ](#).

Casos de uso comuns do ElastiCache e como ele pode ajudar

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

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

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

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

Os exemplos a seguir ilustram algumas das maneiras de como o uso do ElastiCache pode melhorar a performance geral da sua aplicação.

Tópicos

- [Armazenamento de dados na memória](#)
- [Placares de jogos \(conjuntos classificados do Redis\)](#)
- [Mensagens \(publicação/assinatura do Redis\)](#)
- [Dados de recomendação \(hashes do Redis\)](#)
- [Outros usos do Redis](#)
- [Depoimentos de clientes do ElastiCache](#)

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 ver mais informações, consulte:

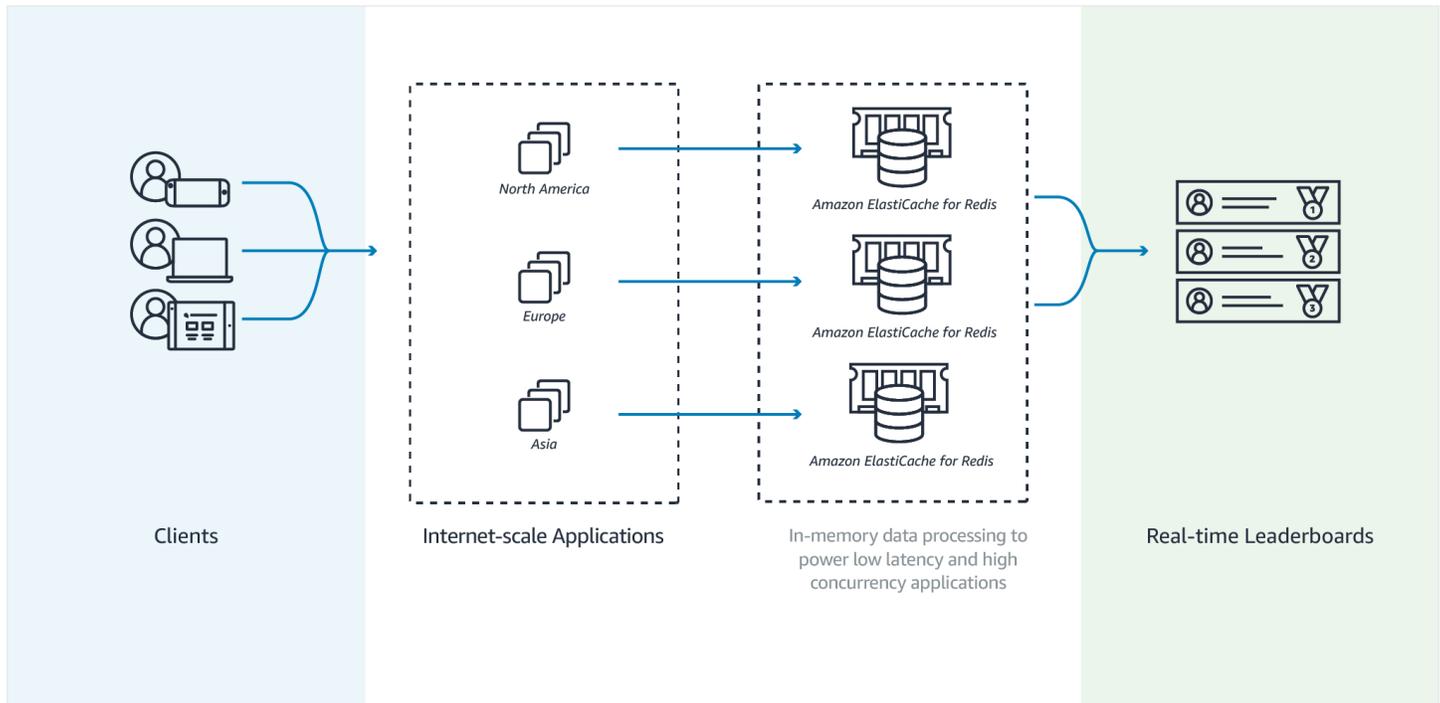
- [Estratégias de armazenamento em cache](#) no Guia do usuário do ElastiCache for Redis

Placares de jogos (conjuntos classificados do Redis)

Conjuntos classificados do Redis movem a complexidade computacional de placares da sua aplicação para o seu cluster do Redis.

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

No diagrama a seguir, veja como funciona um placar de jogos do ElastiCache for Redis.



Example - Placar do Redis

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

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

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

ZADD leaderboard 232 June

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

- 3) Sandra
- 4) Robert

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

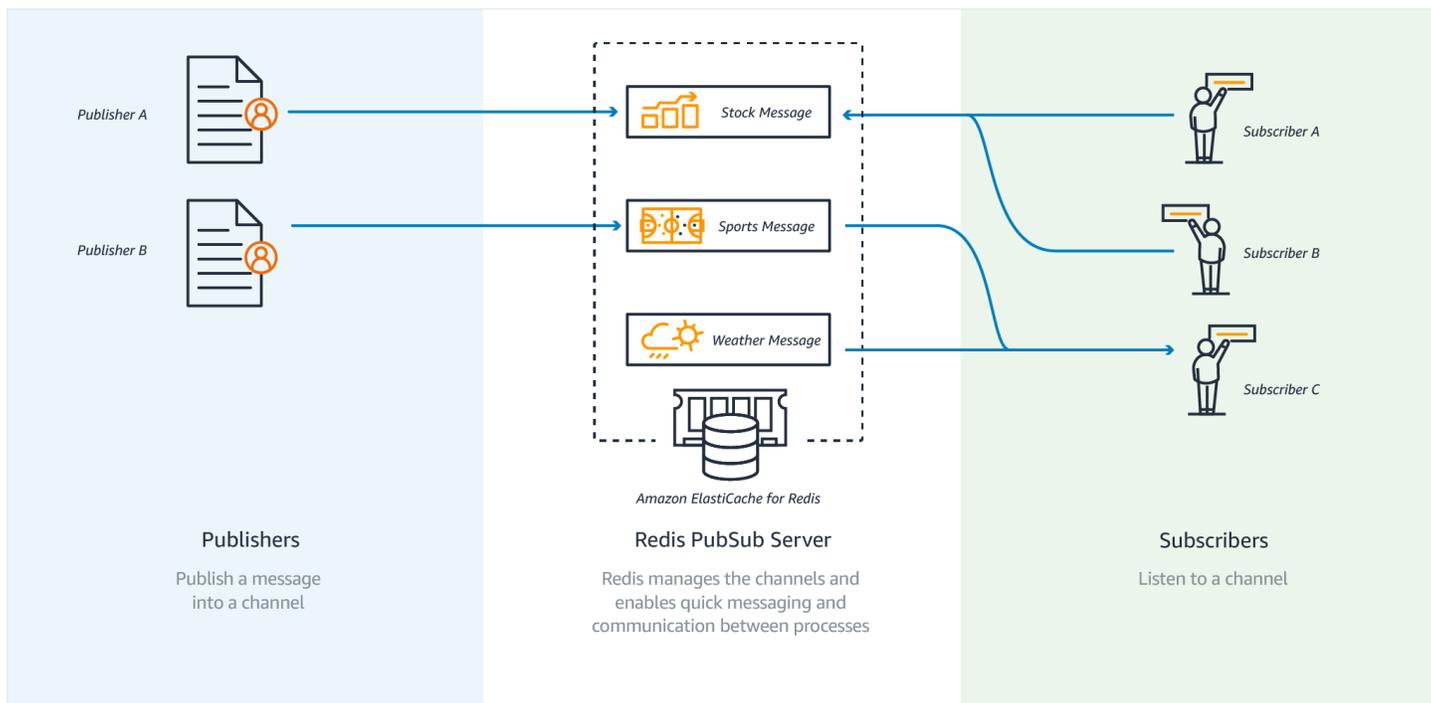
```
ZREVRANK leaderboard June  
1
```

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

Mensagens (publicação/assinatura do Redis)

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

A funcionalidade de publicação/assinatura (pub/sub) do Redis não tem relação com nenhum espaço de chaves. Portanto, ela não interferirá em nenhum nível. No diagrama a seguir há uma ilustração sobre o sistema de mensagens do ElastiCache for Redis.



Assinatura

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

Example - Assinatura de um único canal

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

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

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

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

Example - Assinaturas de vários canais especificados

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

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

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

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

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

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

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

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

ou

```
UNSUBSCRIBE
```

Example - Assinaturas usando a correspondência de padrões

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

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

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

Example Cancelar assinaturas

Para cancelar assinaturas desses canais, use o comando PUNSUBSCRIBE.

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

Important

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

Publicação

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

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

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

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

Dados de recomendação (hashes do Redis)

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

Example - Curtidas e descurtidas

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

Outros usos do Redis

A postagem de blog [How to take advantage of Redis just adding it to your stack](#) (Como tirar proveito do Redis apenas adicionando-o à sua pilha), por Salvatore Sanfilippo, discute uma série de preocupações com banco de dados comuns e como elas podem ser facilmente resolvidas usando o Redis. Essa abordagem elimina a carga do banco de dados e melhora o desempenho.

Depoimentos de clientes do ElastiCache

Para saber mais sobre como empresas como a Airbnb, a PBS a Esri e outras estão usando o Amazon ElastiCache para ampliar seus negócios com uma melhor experiência para o cliente, consulte [Como os outros usam o Amazon ElastiCache](#).

Assista também aos [Vídeos de tutorial](#) para conhecer casos de uso de clientes do ElastiCache adicionais.

Começando a usar o Amazon ElastiCache for Redis

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

Tópicos

- [Configuração](#)
- [Etapa 1: criar um cache](#)
- [Etapa 2: ler e gravar dados no cache](#)
- [Etapa 3: \(opcional\) limpar](#)
- [Próximos Passos](#)
- [Conceitos básicos do ElastiCache e dos SDKs da AWS](#)
- [Tutorial: Configurando uma função Lambda para acessar a Amazon em uma ElastiCache Amazon VPC](#)

Configuração

Para configurar ElastiCache:

Tópicos

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

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. Acesse <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Durante a criação da conta, você vai 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 acesso administrativo a um usuário administrativo](#) e utilize somente o usuário raiz para executar as [tarefas que exigem acesso do usuário raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

Criar um usuário 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 a senha.

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

2. Ative a autenticação multifator (MFA) para o usuário raiz.c

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

Criar um usuário administrativo

1. Habilitar o IAM Identity Center.

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

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

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.

Login como usuário administrativo

- Para fazer login com o usuário do Centro de Identidade do IAM, utilize o URL de login enviado ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

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.

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 (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS	Siga as instruções da interface que deseja utilizar. <ul style="list-style-type: none"> Para o AWS CLI, consulte Configurando o AWS CLI para uso AWS IAM Identity Center no Guia do AWS Command Line Interface usuário.

Qual usuário precisa de acesso programático?	Para	Por
		<ul style="list-style-type: none"> Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de AWS SDKs e ferramentas.
IAM	Use credenciais temporárias para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para AWS SDKs AWS CLI ou APIs. AWS	<p>Siga as instruções da interface que deseja utilizar.</p> <ul style="list-style-type: none"> Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. Para AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de AWS SDKs e ferramentas. Para AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Tópicos relacionados:

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

Configure suas permissões (somente para novos ElastiCache usuários)

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

- Usuários e grupos em 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 no IAM com provedor de identidades:

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

- Usuários do IAM:

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

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

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

Para obter mais informações, consulte:

- [Criar uma política](#) (IAM)

- [Políticas gerenciadas pela AWS para o Amazon ElastiCache](#)
- [Uso de funções vinculadas ao serviço para o Amazon ElastiCache](#)

Configurar o EC2

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

- Se você ainda não tiver uma instância do EC2, saiba como configurar uma instância do EC2 aqui: [Introdução ao EC2](#).
- A instância do EC2 deve estar na mesma VPC e ter as mesmas configurações de grupo de segurança do cache. Por padrão, a Amazon ElastiCache cria um cache na sua VPC padrão e usa o grupo de segurança padrão. Para seguir este tutorial, verifique se a instância do EC2 está na VPC padrão e tem o grupo de segurança padrão.

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

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

1. Faça login no AWS Command Line Interface e abra o console do [Amazon EC2](#).
2. No painel de navegação, em Network & Security, escolha Security Groups.
3. Na lista de grupos de segurança, escolha o de segurança para a sua Amazon VPC. A menos que você tenha criado um grupo de segurança para ElastiCache uso, esse grupo de segurança será denominado padrão.
4. Escolha a guia Entrada e, em seguida:
 - a. Selecione a opção Editar.
 - b. Escolha Adicionar regra.
 - c. Na coluna Tipo, escolha Regra TCP personalizada.
 - d. Na caixa Intervalo de portas, digite 6379.

- e. Na caixa Origem, escolha Qualquer lugar, que tem o intervalo de portas (0.0.0.0/0) para que qualquer instância do Amazon EC2 que você inicie no Amazon VPC possa se conectar ao cache.
- f. Se você estiver usando a tecnologia ElastiCache sem servidor, adicione outra regra escolhendo Adicionar regra.
- g. Na coluna Tipo, escolha Regra TCP personalizada.
- h. Na caixa Intervalo de portas, digite 6380.
- i. Na caixa Origem, escolha Qualquer lugar, que tem o intervalo de portas (0.0.0.0/0) para que qualquer instância do Amazon EC2 que você inicie no Amazon VPC possa se conectar ao cache.
- j. Escolha Salvar

Baixar e configurar o redis-cli

1. Conexão à sua instância do Amazon EC2 com o utilitário de conexão de sua escolha. Para obter instruções sobre como se conectar a uma instância do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon EC2](#).
2. Baixe e instale o utilitário redis-cli executando o comando indicado para a 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
sudo wget http://download.redis.io/redis-stable.tar.gz
sudo tar xvzf redis-stable.tar.gz
cd redis-stable
sudo make BUILD_TLS=yes
```

Note

- Quando você instala o pacote redis6, ele instala o redis6-cli com suporte à criptografia padrão.
- É importante ter suporte de compilação para TLS ao instalar o redis-cli. ElastiCache O Serverless só pode ser acessado quando o TLS está habilitado.
- Se estiver se conectando a um cluster não criptografado, você não precisará da opção `Build_TLS=yes`.

Etapa 1: criar um cache

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

AWS Management Console

Para criar um novo cache usando o console do ElastiCache:

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

AWS CLI

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

Linux

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

Windows

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

O valor do campo Status está definido como CREATING.

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

Linux

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

Windows

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

Depois de criar o novo cache, avance até [Etapa 2: ler e gravar dados no cache](#).

Etapa 2: ler e gravar dados no cache

Esta seção supõe que você tenha criado uma instância do Amazon EC2 e possa conectar-se a ela. Para obter instruções sobre como fazer isso, consulte o [Guia de conceitos básicos do Amazon EC2](#).

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

Encontre o endpoint do cache

AWS Management Console

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

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, no lado esquerdo do console, escolha Caches do Redis.
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 exemplo da AWS CLI a seguir mostra como encontrar o endpoint para o novo cache usando o comando `describe-serverless-caches`. Depois de executar o comando, procure o campo "Endpoint".

Linux

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

Windows

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

Conectar ao cache do Redis (Linux)

Agora que você possui o endpoint necessário, você pode fazer login em uma instância do EC2 e se conectar ao cache. No exemplo a seguir, você usa o utilitário `redis-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/redis-cli -h cache-endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

Conectar ao cache do Redis (Windows)

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

```
c:\Redis>redis-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 [Etapa 3: \(opcional\) limpar](#).

Etapa 3: (opcional) limpar

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

AWS Management Console

Para excluir o cache usando o console:

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

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

AWS CLI

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

Linux

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

Windows

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

O valor do campo Status está definido como EXCLUSÃO.

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

Próximos Passos

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

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

Conceitos básicos do ElastiCache e dos SDKs da AWS

Esta seção contém tutoriais práticos úteis para ajudar você a saber mais sobre o Amazon ElastiCache. Encorajamos você a trabalhar em um dos tutoriais de linguagem específica.

Note

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

Python e ElastiCache

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

- Criar clusters do ElastiCache (modo cluster habilitado e modo cluster desabilitado)
- Verificar se há usuários ou grupos de usuários; caso contrário, criá-los (apenas Redis 6.0 em diante)

- Conectar ao 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 com este tutorial, é possível consultar a documentação do SDK da AWS para Python (Boto). A seção a seguir é específica do ElastiCache: [cliente de nível baixo do ElastiCache](#)

Pré-requisitos do tutorial

- Configurar uma chave de acesso da AWS para usar os SDKs da AWS. Para obter mais informações, consulte [Configuração](#).
- 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.

Criação de ElastiCache clusters e usuários

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

Tópicos

- [Criar um cluster com modo cluster desabilitado](#)
- [Criar um cluster com modo cluster desabilitado com TLS e RBAC](#)
- [Criar um cluster com modo cluster habilitado](#)
- [Criar um cluster com modo cluster habilitado com TLS e RBAC](#)
- [Verifique se 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,ReplicationGroupDescription='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/red-ug/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='redis',
```

```
        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, Redis 6, one primary and two replicas
```

```
    elasticacheResponse = create_cluster_mode_disabled(
```

```
        #CacheNodeType='cache.m6g.large',
```

```
        EngineVersion='6.0',
```

```
    NumCacheClusters=3,
    ReplicationGroupDescription='Redis cluster mode disabled with replicas',
    ReplicationGroupId='redis202104053'
)

logging.info(elasticacheResponse)
```

Para executar o programa, digite o comando a seguir:

```
python CreateClusterModeDisabledCluster.py
```

Para ter mais informações, consulte [Gerenciamento de clusters](#).

Criar um cluster com modo cluster desabilitado com TLS e RBAC

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

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

```
import boto3
import logging

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

def
create_cluster_mode_disabled_rbac(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumCacheC
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/red-ug/
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': 'redis',
        '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, Redis 6, 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='6.0',
    NumCacheClusters=3,
    ReplicationGroupDescription='Redis cluster mode disabled with replicas',
    ReplicationGroupId='redis202104',
    UserGroupIds=[
        'mygroup'
    ],
    SecurityGroupIds=[
        'sg-7cc73803'
    ],
    CacheSubnetGroupName='default'
)

logging.info(response)

```

Para executar o programa, digite o comando a seguir:

```
python ClusterModeDisabledWithRBAC.py
```

Para ter mais informações, consulte [Gerenciamento de clusters](#).

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/red-ug/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='redis',
    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.redis6.0.cluster.on'
)

return response

# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Redis cluster mode enabled with replicas',
    ReplicationGroupId='redis20210',
    # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
node (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 ter mais informações, consulte [Gerenciamento de clusters](#).

Criar um cluster com modo cluster habilitado com TLS e RBAC

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

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

```
import boto3  
import logging  
  
logging.basicConfig(level=logging.INFO)  
client = boto3.client('elasticache')  
  
def  
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,  
    ReplicationGroupDescription='Sample cache with cluster  
mode enabled', ReplicationGroupId=None, UserGroupIds=None,  
    SecurityGroupIds=None, CacheSubnetGroupName=None, CacheParameterGroupName='default.redis6.0.clus  
    """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/red-ug/  
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': 'redis',
        '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='6.0',
ReplicationGroupDescription='Redis cluster mode enabled with replicas',
ReplicationGroupId='redis2021',
# 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 ter mais informações, consulte [Gerenciamento de clusters](#).

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

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

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

```
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. For
    details refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.RBAC.html#Access-string
    :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
```

Conexão com o ElastiCache

Os exemplos a seguir usam o cliente Redis para se conectar ao 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 e cole o programa a seguir 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 e cole o programa a seguir em um arquivo chamado ConnectClusterModeEnabled.py.

```
from rediscluster import RedisCluster
import logging

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

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

Para executar o programa, digite o comando a seguir:

```
python ConnectClusterModeEnabled.py
```

Exemplos de uso

Os exemplos a seguir usam o SDK boto3 para ElastiCache para operação com o ElastiCache.

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

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

logging.basicConfig(level=logging.INFO)

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

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

def readMessage(groupName=None,streamName=None):
    """Starts a loop reading from 'streamName'
    Multiple threads will read from the same stream consumer group. Consumer group is
    used to coordinate data distribution.
```

Once a thread 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 where 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 sucessfully
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 uma função Lambda para acessar a Amazon em uma ElastiCache Amazon 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 sem servidor](#)
- [Etapa 2: criar uma função do Lambda](#)
- [Etapa 3: testar a função do Lambda](#)
- [Etapa 4: limpar \(opcional\)](#)

Etapa 1: criar um cache sem servidor

Para criar um cache sem servidor, siga estas etapas.

Tópicos

- [Etapa 1.1: Criar um cache sem servidor](#)
- [Etapa 1.2: Copiar o endpoint de cache sem servidor](#)

- [Etapa 1.3: criar a função do IAM](#)
- [Etapa 1.4: Criar um cache sem servidor](#)

Etapa 1.1: Criar um cache sem servidor

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

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

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 o Redis 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 a função do IAM

1. Crie um documento de política de confiança do IAM, conforme mostrado abaixo, para o perfil que permita que sua conta assumo o novo perfil. 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 de política do IAM, 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. Criar um perfil do IAM.

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

4. Crie a política do IAM.

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

5. Anexe a política do IAM à função.

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

Etapa 1.4: Criar um 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 habilitado para o IAM.

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

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

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

Etapa 2: criar uma função do Lambda

Para criar uma função Lambda, siga estas etapas.

Tópicos

- [Etapa 2.1: Criar uma função Lambda](#)
- [Etapa 2.2: Criar o perfil do IAM \(perfil de execução\)](#)
- [Etapa 2.3: fazer upload do pacote de implantação \(criar função do Lambda\)](#)

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 1.2.

```
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 Redis.")
else:
    raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

return "Fetched value from Redis"
```

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 IAM Auth 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 o perfil do IAM (perfil de execução)

Anexe a política AWS gerenciada nomeada `AWSLambdaVPCAccessExecutionRole` à função.

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

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

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

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

Para a opção de função, use o ARN da função de execução que você criou na etapa 2.2. Para o `vpc-config`, insira listas separadas por vírgulas das sub-redes da VPC padrão e o ID do grupo de segurança da VPC padrão. É possível encontrar esses valores no console do Amazon VPC. Para encontrar as sub-redes da sua VPC padrão, escolha Suas VPCs e, em seguida, escolha a VPC padrão da sua conta AWS. Para encontrar o grupo de segurança dessa VPC, acesse Segurança e escolha Grupos de segurança. Não se esqueça de selecionar a região `us-east-1`.

```
aws lambda create-function \  
  --function-name AccessRedis \  
  --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: testar a função do Lambda

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 `AccessRedis` () usando AWS Lambda o comando `invoke`.

```
aws lambda invoke \  
--function-name AccessRedis \  
--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 (AccessRedis/aws/lambda/). O fluxo de logs deve conter uma saída semelhante à mostrada a seguir:

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

- Analise os resultados no AWS Lambda console.

Etapa 4: limpar (opcional)

Para limpar, siga estas etapas.

Tópicos

- [Etapa 4.1: Excluir a função Lambda](#)
- [Etapa 4.2: Excluir o cache sem servidor](#)
- [Etapa 4.3: Remover a função e as políticas do IAM](#)

Etapa 4.1: Excluir a função Lambda

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

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 a função e as políticas do IAM

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

Projeto e gerenciamento do próprio cluster do ElastiCache

Se precisar de controle refinado sobre o cluster do ElastiCache, você poderá optar por criar o próprio cluster. O ElastiCache permite a você operar um cluster baseado em nó escolhendo o tipo de nó, o número de nós e o posicionamento do nó nas zonas de disponibilidade do cluster AWS. Por ser um serviço totalmente gerenciado, o ElastiCache gerencia automaticamente o provisionamento, o monitoramento, as substituições de nó e a aplicação de patches de software para o cluster.

Para obter informações sobre como configurar, consulte [Configuração](#). Para obter detalhes sobre como gerenciar, atualizar ou excluir nós ou clusters, consulte [Gerenciamento de nós](#). Para obter uma visão geral dos componentes principais de uma implantação do Amazon ElastiCache ao projetar o próprio cluster do ElastiCache, consulte os conceitos principais.

Tópicos

- [ElastiCache para componentes e recursos do Redis](#)
- [Terminologia do ElastiCache for Redis](#)
- [Projeto do próprio cluster](#)
- [Gerenciamento de nós](#)
- [Gerenciamento de clusters](#)
- [Comparação dos caches autoprojetados do Memcached e do Redis](#)
- [Migração online para o ElastiCache](#)
- [Escolher regiões e zonas de disponibilidade](#)

ElastiCache para componentes e recursos do Redis

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

Tópicos

- [ElastiCache nós](#)
- [ElastiCache para fragmentos do Redis](#)
- [ElastiCache para clusters Redis](#)
- [ElastiCache para replicação do Redis](#)
- [AWS Regiões e zonas de disponibilidade](#)

- [ElastiCache para endpoints Redis](#)
- [ElastiCache grupos de parâmetros](#)
- [ElastiCache para segurança do Redis](#)
- [ElastiCache grupos de sub-redes](#)
- [ElastiCache para backups do Redis](#)
- [ElastiCache eventos](#)

ElastiCache nós

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

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

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

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

Para obter mais informações sobre nós, consulte [Gerenciamento de nós](#).

ElastiCache para fragmentos do Redis

Um fragmento do Redis (chamado de grupo de nós na API e na CLI) é um agrupamento de um a seis nós relacionados. Um cluster Redis (modo de cluster desativado) 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.

Clusters do Redis (modo cluster habilitado) podem ter até 500 fragmentos, com seus dados particionados nos fragmentos. O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do 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 as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para ter 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 ter mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

Para obter mais informações sobre estilhaços, consulte [Operação com fragmentos](#).

ElastiCache para clusters Redis

Um cluster Redis é um agrupamento lógico de um ou mais [ElastiCache para fragmentos do Redis](#). Os dados são particionados em todos os fragmentos em um cluster do Redis (modo cluster habilitado).

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:

- [Gerenciamento de clusters](#) e [Gerenciamento de nós](#)

Informações sobre clusters, nós e operações relacionadas.

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

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

- [Atenuar falhas](#)

Informações sobre como melhorar a tolerância a falhas dos seus clusters e grupos de replicação.

Configurações de cluster típicas

Veja a seguir configurações de cluster típicas.

Clusters do Redis

Clusters do Redis (modo cluster desabilitado) sempre contêm apenas um fragmento (na API e CLI, um grupo de nós). Um fragmento do 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 cluster do Redis e habilitar o Multi-AZ. Para ter mais informações, consulte [Atenuar falhas](#).

À medida que a demanda sobre o seu cluster do Redis (modo cluster desabilitado) mudar, você poderá expandir ou reduzir a escala. Para fazer isso, mova seu cluster para um tipo de instância de nó diferente. Se a sua aplicação exigir leituras intensas, recomendamos adicionar um cluster do Redis (modo cluster desabilitado) de réplica somente de leitura. 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 ter mais informações, consulte [Classificação de dados em níveis](#).

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

ElastiCache para replicação do Redis

A replicação é implementada agrupando de dois a seis nós em um fragmento (na API e CLI, chamado de grupo de nós). Um desses nós é o nó primário de leitura/gravação. Todos os outros nós são nós de réplica somente leitura.

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](#).

Clusters do Redis (modo cluster desabilitado) oferecem suporte a um fragmento (na API e CLI, um grupo de nós).

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

Comparando replicação: Redis (modo cluster desabilitado) e Redis (modo cluster habilitado)

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

	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
Fragmentos (grupos de nós)	1	1–500

	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
Réplicas para cada fragmento (grupo de nós)	0–5	0–5
Particionamento de dados	Não	Sim
Adiciona/Exclui réplicas	Sim	Sim
Adiciona/Exclui grupos de nós	Não	Sim
Suporte a aumento de escala na horizontal	Sim	Sim
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 o backup e a restauração para migrar para Redis (modo cluster habilitado).

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

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

Réplicas de leitura protegem contra a perda de dados em potencial, pois seus dados são replicados em dois ou mais nós: o primário e uma ou mais réplicas de leitura. Para maior confiabilidade

e recuperação mais rápida, recomendamos que você crie uma ou mais réplicas de leitura em diferentes zonas de disponibilidade.

Você também pode aproveitar os datastores globais. Ao usar o recurso Global Datastore for Redis, 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 para o Redis 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

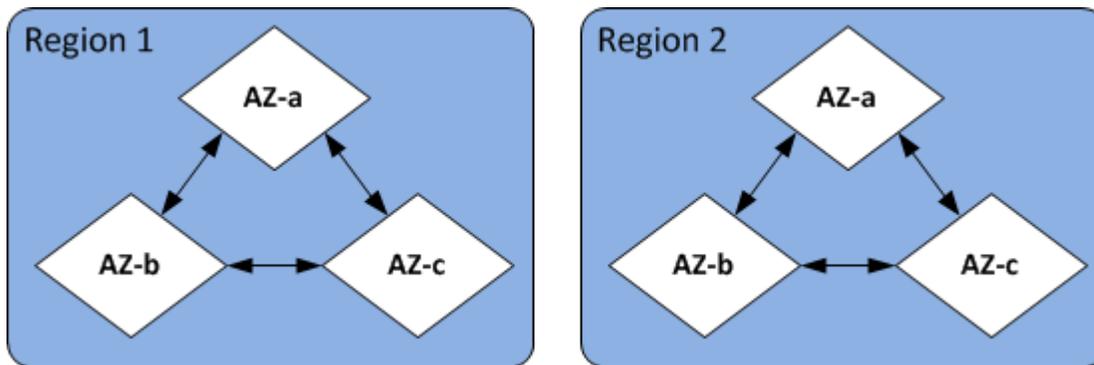
- O Multi-AZ não tem suporte em tipos de nó T1.

AWS Regiões e zonas de disponibilidade

A Amazon ElastiCache está disponível em várias AWS regiões ao redor do mundo. Assim, você pode lançar ElastiCache clusters em locais que atendam às suas necessidades de negócios. Por exemplo, você pode lançar na AWS região mais próxima de seus clientes ou para atender a determinados requisitos legais.

Por padrão, os AWS SDKs AWS CLI, a ElastiCache API e o ElastiCache console fazem referência à região Oeste dos EUA (Oregon). À medida que a disponibilidade ElastiCache se expande para novas AWS regiões, novos endpoints para essas AWS regiões também estão disponíveis. Você pode usá-los em suas solicitações HTTP, nos AWS SDKs e no ElastiCache console. AWS CLI

Cada AWS região foi projetada para ser completamente isolada das outras AWS regiões. Dentro de cada região, há várias zonas de disponibilidade. Ao iniciar seus nós em diferentes zonas de disponibilidade, você é capaz de alcançar o máximo possível de tolerância a falhas. Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Escolher regiões e zonas de disponibilidade](#). No diagrama a seguir, você pode ver uma visão geral de como AWS as regiões e as zonas de disponibilidade funcionam.



Para obter informações sobre AWS regiões suportadas por ElastiCache e seus endpoints, consulte [Regiões e endpoints com suporte](#).

ElastiCache para endpoints Redis

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

Endpoint de nó único para Redis (modo cluster desabilitado)

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

Endpoint de vários nós para Redis (modo cluster desabilitado)

Um cluster do Redis (modo cluster desabilitado) de vários nós 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 Endpoints de nó individuais para operações de leitura (na API/CLI, eles são chamados de Endpoints de leitura).

Endpoints do Redis (modo cluster ativado)

O cluster do Redis (modo cluster habilitado) possui um único endpoint de configuração. Ao se conectar ao endpoint de configuração, a aplicação pode descobrir os endpoints primários e os endpoints de leitura para cada fragmento no cluster.

Para ter mais informações, consulte [Localização de endpoints de conexão](#).

ElastiCache grupos de parâmetros

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

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

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

ElastiCache para segurança do Redis

Para maior segurança, ElastiCache para Redis, o acesso ao nó é restrito aos aplicativos executados nas instâncias do Amazon EC2 que você permite. É possível controlar as instâncias do Amazon EC2 que podem acessar seu cluster usando grupos de segurança.

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

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

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

ElastiCache grupos de sub-redes

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

Se você criar um cluster em uma Amazon VPC, deverá especificar um grupo de sub-rede 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 ambiente da Amazon VPC, consulte o seguinte:

- [Segurança do Amazon VPCs e do ElastiCache](#)
- [Etapa 3: autorizar o acesso ao cluster](#)
- [Sub-redes e grupos de sub-redes](#)

ElastiCache para backups do Redis

Um backup é uma point-in-time cópia de um cluster Redis. 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 Redis em execução no seu cluster, o processo de backup exige diferentes quantidades de memória reservada para ter sucesso. Para mais informações, consulte:

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

ElastiCache eventos

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

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

Terminologia do ElastiCache for Redis

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

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

Cluster ou nó de cache vs. nó

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

A API do ElastiCache e a AWS CLI continuam a usar os termos como antes.

Cluster vs. grupo de replicação

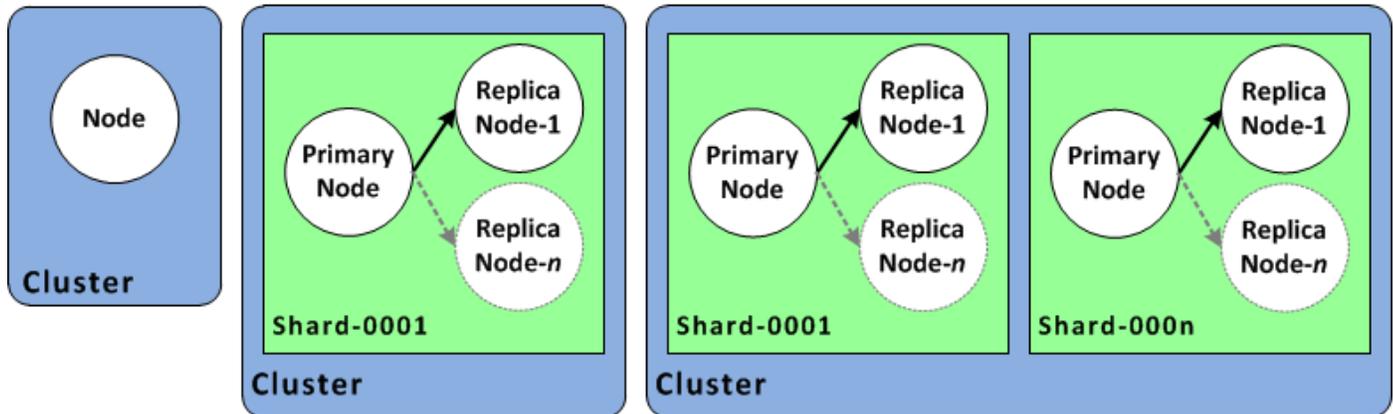
Agora, o console usa o termo cluster para todos os clusters do ElastiCache for Redis. O console usa o termo Cluster em todas essas circunstâncias:

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

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

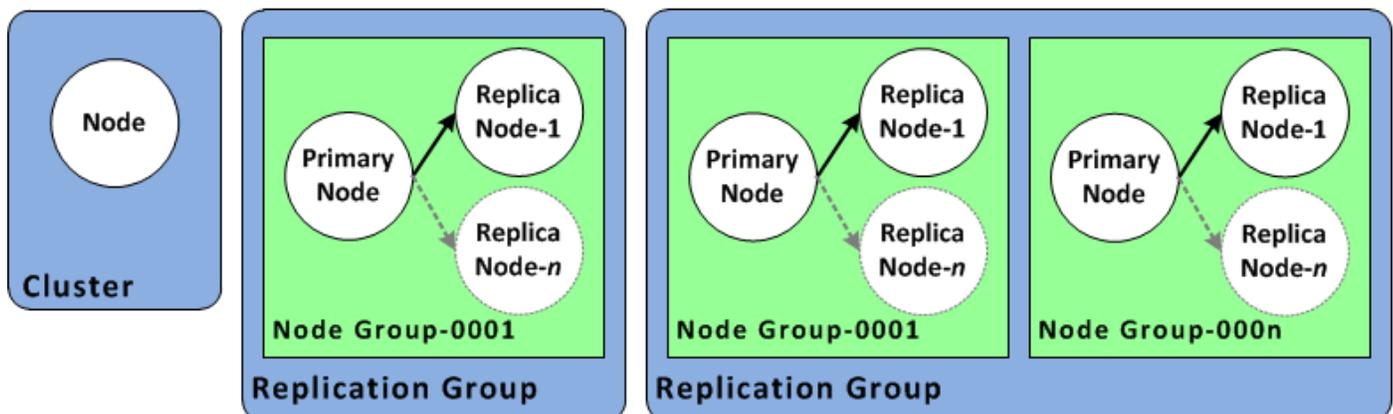
O diagrama a seguir ilustra as várias topologias de clusters do ElastiCache for Redis na perspectiva do console.

ElastiCache for Redis: Console View



As operações da API do ElastiCache e da AWS CLI ainda distinguem clusters do ElastiCache for Redis de nó único de grupos de replicação de vários nós. O diagrama a seguir ilustra as várias topologias do ElastiCache for Redis sob a perspectiva da API do ElastiCache e da AWS CLI.

ElastiCache for Redis: API/CLI View



Grupo de replicação versus datastore global

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

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

Para obter informações sobre armazenamentos de dados globais, consulte [Replicação em regiões da AWS usando o armazenamentos de dados globais](#).

Projeto do próprio cluster

A seguir estão as ações únicas que você deve realizar para começar a projetar seu ElastiCache cluster.

Tópicos

- [Configuração](#)
- [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 5: excluir um cluster](#)
- [Tutoriais e vídeos do ElastiCache](#)
- [O que faço agora?](#)

Configuração

Antes de criar seu cluster, crie primeiro um grupo de sub-redes. Um grupo de sub-redes de cache é uma coleção de sub-redes que você pode querer designar aos seus clusters de cache em uma VPC. Ao executar um cluster de cache em uma VPC, é necessário selecionar um grupo de sub-redes de cache. Em seguida, o ElastiCache usa esse grupo de sub-rede de cache para atribuir endereços IP dentro dessa sub-rede a cada nó de cache do cluster.

Quando você criar um novo grupo de sub-rede, 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 como configurar o ElastiCache, consulte [Configuração](#).

Etapa 1: criar um grupo de sub-redes

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

Subnet group settings

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

Name

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- <input type="text"/>		172.31.16.0/20
us-east-1b	subnet- <input type="text"/>		172.31.32.0/20
us-east-1c	subnet- <input type="text"/>		172.31.0.0/20
us-east-1d	subnet- <input type="text"/>		172.31.80.0/20

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

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

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 tópico sobre a AWS CLI [create-cache-subnet-group](#).

Etapa 2: criar um cluster

Antes de criar um cluster para uso em produção, é óbvio que você precisa considerar como configurar o cluster para atender às suas necessidades de negócios. Esses problemas são abordados na seção [Preparação de um cluster](#). Para os propósitos deste exercício de introdução, você criará um cluster com o modo cluster desabilitado e poderá aceitar os valores de configuração padrão onde eles se aplicam.

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

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

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

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

Para criar um cluster Redis (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 Create VPC (Criar VPC) e siga as etapas em [Creating a Virtual Private Cloud \(VPC\)](#) (Criação de uma nuvem privada virtual (VPC)).
5. Na página do ElastiCache painel, escolha Cache Redis e, em seguida, escolha Criar cache Redis.
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 Redis com ElastiCache 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 específicos do Redis](#) e [Criar um parameter group](#).

 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 ter 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 ter mais informações, consulte [Classificação de dados em níveis](#).

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

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

Para obter mais informações, consulte:

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

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

Para ter 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 ter mais informações, consulte [Escolher regiões e zonas de disponibilidade.](#)

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

 Note

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

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

 Note

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

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

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

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

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

Para ter mais informações, consulte [Gerenciamento da manutenção](#).

9. (Opcional) Para Logs:
 - Em Log format (Formato do log), escolha Text (Texto) ou JSON.
 - Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
 - Em Destino do registro, escolha Criar novo e insira o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose, ou escolha Selecionar existente e escolha o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose,
10. Para Tags, para ajudá-lo a gerenciar seus clusters e outros ElastiCache recursos, você pode atribuir seus próprios metadados a cada recurso na forma de tags. Para obter mais informações, consulte [Marcação de recursos do 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 Redis com ElastiCache 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 for available (disponível), você poderá conceder acesso ao Amazon EC2 a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 4: conexão ao nó do cluster](#) e [Etapa 3: autorizar o acesso ao 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 [Excluir um cluster](#).

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

Example

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

Para Linux, macOS ou Unix:

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

Para Windows:

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

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

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

Etapa 3: autorizar o acesso ao cluster

Esta seção supõe que você esteja familiarizado com a execução e a conexão com instâncias do Amazon EC2. Para obter mais informações, consulte o [Guia de conceitos básicos do Amazon EC2](#).

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

Por padrão, o acesso de rede ao seu cluster é limitado à conta que foi usada para criá-lo. Antes de poder se conectar a um cluster a partir de uma instância do EC2, você deve autorizar a instância do EC2 para acessar o cluster. As etapas necessárias dependem de se você iniciou seu cluster no EC2-VPC ou no EC2-Classic.

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

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

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

Important

Os grupos de segurança do Amazon ElastiCache apenas são aplicáveis a clusters que não estejam 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 nós do ElastiCache em uma Amazon VPC, controlará o acesso aos seus clusters com grupos de segurança da Amazon VPC, que são diferentes de grupos de segurança do ElastiCache. Para obter mais informações sobre como usar o ElastiCache em uma Amazon VPC, consulte [Segurança do Amazon VPCs e do ElastiCache](#)

2. Crie um grupo de segurança de VPC para suas instâncias do EC2 (servidores Web e de aplicativos). Esse grupo de segurança pode, se necessário, permitir o acesso à instância do EC2 da Internet através da tabela de rotas da VPC. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o acesso TCP à instância do EC2 pela porta 22.
3. Crie regras personalizadas no security group para o seu cluster que permitam conexões do security group que você criou para suas instâncias do EC2. Isso permitiria que qualquer membro de grupo de segurança acessasse os clusters.

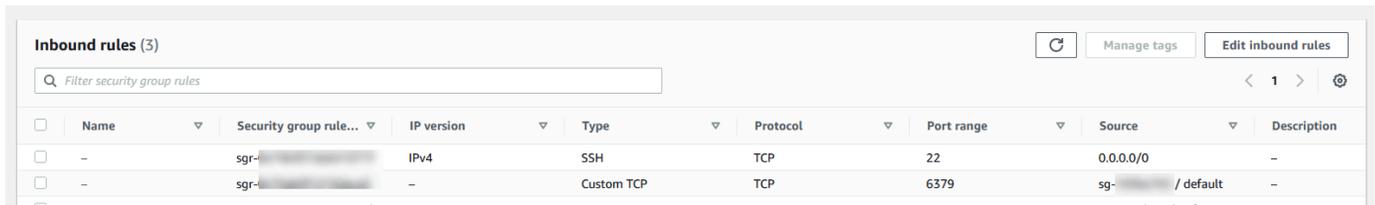
 Note

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

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

1. Faça login no Console de Gerenciamento da AWS e abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc>.
2. No painel de navegação, selecione 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 Type, escolha Custom TCP Rule.
 - a. Para Port Range, especifique a porta que você usou quando criou seu cluster.

A porta padrão dos clusters e grupos de replicação 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 o suas instâncias do Amazon EC2.
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 cluster do ElastiCache de uma Amazon VPC diferente, uma região da AWS diferente ou até mesmo sua rede corporativa, consulte:

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

Etapa 4: conexão ao nó do cluster

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

Esta seção supõe que você tenha criado uma instância do Amazon EC2 e possa conectar-se a ela. Para obter instruções sobre como fazer isso, consulte o [Guia de conceitos básicos do Amazon EC2](#).

Uma instância do Amazon EC2 pode se conectar a um nó de cluster apenas se você tiver a autorização a fazê-lo.

Localize seus endpoints de nó

Quando seu cluster estiver no estado available (disponível) e você tiver autorizado o acesso a ele, será possível fazer login em uma instância do Amazon EC2 e se conectar a ela. Para isso, primeiro você deve determinar o endpoint.

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

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

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

Um endpoint de leitor divide uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura em um cluster do ElastiCache para Redis. Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar várias réplicas de leitura do seu cluster do ElastiCache para Redis em diferentes zonas de disponibilidade da AWS (AZ) para garantir a alta disponibilidade dos endpoints leitores.

Note

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

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

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

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

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

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

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

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

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

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

<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

Endpoint de nó para um cluster do Redis (modo cluster desabilitado)

5. Como copiar um endpoint para a área de transferência:
 - a. Um endpoint por vez, localize o endpoint que você deseja copiar.
 - b. Escolha o ícone de cópia diretamente na frente do endpoint.

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

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

Criptografia em trânsito não habilitada

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

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

Criptografia em trânsito habilitada

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

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

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

- [Localização de endpoints de conexão](#)
- [Localização de endpoints para um cluster do Redis \(modo cluster habilitado\) \(console\)](#): você precisa do endpoint de configuração do cluster.

- [Localizar endpoints \(AWS CLI\)](#)
- [Localização de endpoints \(API do ElastiCache\)](#)

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

Agora que você tem o endpoint necessário, você pode fazer login em uma instância do EC2 e se conectar ao cluster ou ao grupo de replicação. No exemplo a seguir, você usa o utilitário `redis-cli` para se conectar a um cluster. A versão mais recente da `redis-cli` também oferece suporte a SSL/TLS para conexão com clusters habilitados para criptografia e autenticação.

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

 Note

Este processo abrange o teste de uma conexão usando o utilitário `redis-cli` somente para uso não planejado. Para obter uma lista dos clientes de Redis compatíveis, consulte a [Documentação do Redis](#). Para obter exemplos de como usar os SDKs da AWS com ElastiCache, consulte [Conceitos básicos do ElastiCache e dos SDKs da AWS](#).

Conexão a um cluster não criptografado com modo cluster desabilitado

1. Execute o seguinte comando para se conectar ao cluster e substitua *primary-endpoint* e *port-number* pelo endpoint do seu cluster e pelo número da porta. (A porta padrão para o Redis é 6379.)

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

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

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

2. Agora você pode executar comandos do Redis.

```
set x Hello  
OK
```

```
get x
"Hello"
```

Conectar a um cluster não criptografado com modo cluster habilitado

1. Execute o seguinte comando para se conectar ao cluster e substitua *configuration-endpoint* e *port-number* pelo endpoint do seu cluster e pelo número da porta. (A porta padrão para o Redis é 6379.)

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

 Note

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

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

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

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

```
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, a `redis-cli` usa uma conexão TCP não criptografada ao se conectar ao Redis. A opção `BUILD_TLS=yes` habilita o SSL/TLS no momento da compilação da `redis-cli`, conforme mostrado na seção [Baixar e configurar o redis-cli](#). A ativação do AUTH é opcional. No entanto, você deve habilitar a criptografia em trânsito para habilitar o AUTH. Para obter mais detalhes sobre criptografia e autenticação do ElastiCache, consulte [Criptografia em trânsito \(TLS\) do ElastiCache](#).

Note

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

Nos exemplos a seguir, substitua *cluster-endpoint* e *port-number* pelo endpoint do seu cluster e seu número de porta. (A porta padrão para o Redis é 6379.)

Conexão a clusters criptografados com modo cluster desabilitado

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

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

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

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

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

Alternativa com redis-cli

Se o cluster não estiver habilitado para o modo cluster e você precisar fazer uma conexão com o cluster para um teste curto, mas sem passar pela compilação da redis-cli, você pode usar telnet ou openssl. Nos exemplos de comandos a seguir, substitua *cluster-endpoint* e *port-number* pelo endpoint do seu cluster e seu número de porta. (A porta padrão para o Redis 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* pela senha do seu cluster.

```
Auth your-password
```

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

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

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

Para se conectar ao cluster do Redis a partir de uma instância do EC2 do Windows usando a CLI do Redis, é necessário fazer download do pacote redis-cli e usar redis-cli.exe para se conectar ao cluster do Redis de uma instância do EC2 do Windows.

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

Para se conectar a um cluster do Redis não habilitado para criptografia usando o redis-cli

1. Conexão à sua instância do Amazon EC2 com o utilitário de conexão de sua escolha. Para obter instruções sobre como se conectar a uma instância do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon EC2](#).
2. Copie e cole o link <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> em um navegador da Internet para baixar o arquivo zip para o cliente Redis da versão disponível no GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extraia o arquivo zip para a pasta/caminho desejado.

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

Por exemplo:

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

3. Execute os comandos Redis.

Agora, você está conectado ao cluster e pode executar comandos Redis como os abaixo.

```
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 redis-cli
```

Etapa 5: 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

Ao excluir um cluster do ElastiCache for Redis, seus snapshots 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](#).

Usar a 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 console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel do console do ElastiCache, escolha Redis.

Uma lista de todos os caches que executam o Redis 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 de cada vez no console do ElastiCache. 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.

Usar a 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 ação `delete-cache-cluster` da CLI apenas exclui 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 a AWS CLI para o tópico do ElastiCache [delete-cache-cluster](#).

Tutoriais e vídeos do ElastiCache

Os seguintes tutoriais discutem tarefas de interesse para o usuário do Amazon ElastiCache.

- [Vídeos do ElastiCache](#)
- [Tutorial: configurar uma função Lambda para acessar o Amazon ElastiCache em uma Amazon VPC](#)

Vídeos do ElastiCache

A seguir, você pode encontrar vídeos para ajudá-lo a aprender os conceitos básicos e avançados do Amazon ElastiCache. Para obter mais informações sobre treinamento da AWS, consulte [Treinamento e certificação da AWS](#).

Tópicos

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

Vídeos de introdução

Os seguintes vídeos apresentam você ao Amazon ElastiCache.

Tópicos

- [AWS re:Invent 2020: o que há de novo no Amazon ElastiCache](#)
- [AWS re:Invent 2019: o que há de novo no Amazon ElastiCache](#)
- [AWS re:Invent 2017: o que há de novo no Amazon ElastiCache](#)
- [DAT204: Criação de aplicações escaláveis em serviços NoSQL da AWS \(re:Invent 2015\)](#)
- [DAT207: Aceleração de performance da aplicação com o Amazon ElastiCache \(AWS re:Invent 2013\)](#)

AWS re:Invent 2020: o que há de novo no Amazon ElastiCache

[AWS re:Invent 2020: o que há de novo no Amazon ElastiCache](#)

AWS re:Invent 2019: o que há de novo no Amazon ElastiCache

[AWS re:Invent 2019: o que há de novo no Amazon ElastiCache](#)

AWS re:Invent 2017: o que há de novo no Amazon ElastiCache

[AWS re:Invent 2017: o que há de novo no Amazon ElastiCache](#)

DAT204: Criação de aplicações escaláveis em serviços NoSQL da AWS (re:Invent 2015)

Nesta sessão, discutimos os benefícios dos bancos de dados NoSQL e fazemos um tour pelos principais serviços NoSQL oferecidos pela AWS: Amazon DynamoDB e Amazon ElastiCache. Em

seguida, ouvimos dois clientes líderes, a Expedia e a Mapbox, sobre seus casos de uso e desafios arquitetônicos e como eles os enfrentaram usando os serviços NoSQL da AWS, incluindo padrões de design e melhores práticas. Você sairá dessa sessão com uma melhor compreensão sobre o NoSQL e seus poderosos recursos, pronto para enfrentar seus desafios de banco de dados com confiança.

[DAT204: Criação de aplicações escaláveis em serviços NoSQL da AWS \(re:Invent 2015\)](#)

DAT207: Aceleração de performance da aplicação com o Amazon ElastiCache (AWS re:Invent 2013)

Neste vídeo, saiba como você pode usar o Amazon ElastiCache para implantar facilmente um sistema de cache de memória para acelerar a performance da aplicação. Mostramos como usar o Amazon ElastiCache para melhorar a latência da aplicação e reduzir a carga nos seus servidores de banco de dados. Também mostramos como criar uma camada de cache que é fácil de gerenciar e escalar à medida que o seu aplicativo se expande. Durante esta sessão, examinamos vários cenários e casos de uso que podem se beneficiar com a habilitação do armazenamento em cache, além de discutirmos os recursos fornecidos pelo Amazon ElastiCache.

[DAT207: Aceleração de performance de aplicação com o Amazon ElastiCache \(re:Invent 2013\)](#)

Vídeos avançados

Os seguintes vídeos abrangem tópicos mais avançados do Amazon ElastiCache.

Tópicos

- [Design para o sucesso com as práticas recomendadas do Amazon ElastiCache \(re:Invent 2020\)](#)
- [Sobrecarregue suas aplicações em tempo real com o Amazon ElastiCache \(re:Invent 2019\)](#)
- [Práticas recomendadas: migração de clusters Redis do Amazon EC2 para o ElastiCache \(re:Invent 2019\)](#)
- [Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Redis confiável e escalável na nuvem com o Amazon ElastiCache \(re:Invent 2018\)](#)
- [Aprofundamento no ElastiCache: padrões de design para armazenamentos de dados na memória \(re:Invent 2018\)](#)
- [DAT305: Aprofundamento no Amazon ElastiCache \(re:Invent 2017\)](#)
- [DAT306: Aprofundamento no Amazon ElastiCache \(re:Invent 2016\)](#)
- [DAT317: Como o IFTTT usa o ElastiCache for Redis a fim de prever eventos \(re:Invent 2016\)](#)

- [DAT407: Aprofundamento no Amazon ElastiCache \(re:Invent 2015\)](#)
- [SDD402: Aprofundamento no Amazon ElastiCache \(re:Invent 2014\)](#)
- [DAT307: Aprofundamento na arquitetura e nos padrões de design do Amazon ElastiCache \(re:Invent 2013\).](#)

Design para o sucesso com as práticas recomendadas do Amazon ElastiCache (re:Invent 2020)

Com o crescimento explosivo de aplicações críticas para os negócios e em tempo real criadas com base no Redis, a disponibilidade, a escalabilidade e a segurança se tornaram as principais considerações. Conheça as melhores práticas para configurar o Amazon ElastiCache para obter sucesso com escalabilidade on-line, alta disponibilidade em implantações multi-AZ e configurações de segurança.

[Design para o sucesso com as práticas recomendadas do Amazon ElastiCache \(re:Invent 2020\)](#)

Sobrecarregue suas aplicações em tempo real com o Amazon ElastiCache (re:Invent 2019)

Com o rápido crescimento da adoção da nuvem e os novos cenários que ela capacita, as aplicações precisam de latência de microssegundos e alto throughput para suportar milhões de solicitações por segundo. Tradicionalmente, os desenvolvedores contam com hardware especializado e soluções alternativas, como bancos de dados baseados em disco combinados com técnicas de redução de dados, para gerenciar dados para aplicações em tempo real. Essas abordagens podem ser caras e não escaláveis. Saiba como aumentar a performance de aplicações em tempo real usando o Amazon ElastiCache totalmente gerenciado e na memória para obter performance extrema, alta escalabilidade, disponibilidade e segurança.

[Sobrecarregue suas aplicações em tempo real com o Amazon ElastiCache \(re:Invent 2019:\)](#)

Práticas recomendadas: migração de clusters Redis do Amazon EC2 para o ElastiCache (re:Invent 2019)

Gerenciar clusters Redis por conta própria pode ser difícil. Você precisa provisionar hardware, corrigir software, fazer backup de dados e monitorar cargas de trabalho constantemente. Com o recém-lançado recurso de migração online para o Amazon ElastiCache, agora você pode mover facilmente seus dados do Redis auto-hospedado no Amazon EC2 para o Amazon ElastiCache totalmente gerenciado, com o modo cluster desabilitado. Nesta sessão, você aprenderá sobre a nova ferramenta de migração online, verá uma demonstração e, mais importante, aprenderá práticas recomendadas para uma migração tranquila para o Amazon ElastiCache.

[Práticas recomendadas: migração de clusters Redis do Amazon EC2 para o ElastiCache \(re:Invent 2019\)](#)

Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora STP11 (re:Invent 2018)

Dream11 é a principal startup de tecnologia esportiva da Índia. Ela tem uma base crescente de mais de 40 milhões de usuários jogando vários esportes, incluindo críquete, futebol e basquete de fantasia, e atualmente atende a um milhão de usuários simultâneos, que produzem três milhões de solicitações por minuto sob um tempo de resposta de 50 milissegundos. Nesta palestra, o CTO da Dream11, Amit Sharma, explica como a empresa usa o Amazon Aurora e o Amazon ElastiCache para lidar com o tráfego instantâneo, que pode triplicar em uma janela de resposta de 30 segundos. Sharma também fala sobre escalar transações sem bloquear, e ele compartilha as etapas para lidar com o tráfego instantâneo, servindo assim a cinco milhões de usuários ativos diários. Título completo: AWS re:Invent 2018: Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora (STP11)

[Escalabilidade de uma plataforma de esportes de fantasia com o Amazon ElastiCache e o Amazon Aurora \(STP11\) \(re:Invent 2018\)](#)

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

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

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

Aprofundamento no ElastiCache: padrões de design para armazenamentos de dados na memória (re:Invent 2018)

Nesta sessão, fornecemos uma visão dos bastidores para saber mais sobre o design e a arquitetura do Amazon ElastiCache. Veja os padrões de design comuns com nossas ofertas do Redis e do Memcached e como os clientes os usam para processamento de dados na memória para reduzir a latência e melhorar o throughput da aplicação. Revisamos as melhores práticas, os padrões de design e os antipadrões do ElastiCache.

[Aprofundamento no ElastiCache: padrões de design para armazenamentos de dados na memória \(re:Invent 2018\)](#)

DAT305: Aprofundamento no Amazon ElastiCache (re:Invent 2017)

Confira os bastidores para saber mais sobre o design e a arquitetura da Amazon ElastiCache. Veja os padrões de design comuns com nossas ofertas do Memcached e do Redis e como os clientes os usaram para operações na memória para reduzir a latência e melhorar o throughput do aplicativo. Neste vídeo, revisamos as melhores práticas, os padrões de design e os antipadrões do ElastiCache.

O vídeo apresenta o seguinte:

- Refragmentação online do ElastiCache for Redis
- Segurança e criptografia do ElastiCache
- ElastiCache for Redis versão 3.2.10

[DAT305: Aprofundamento no Amazon ElastiCache \(re:Invent 2017\)](#)

DAT306: Aprofundamento no Amazon ElastiCache (re:Invent 2016)

Confira os bastidores para saber mais sobre o design e a arquitetura da Amazon ElastiCache. Veja os padrões de design comuns com nossas ofertas do Memcached e do Redis e como os clientes os usaram para operações na memória para reduzir a latência e melhorar o throughput do aplicativo. Durante esta sessão, analisamos as melhores práticas, os padrões de design e os antipadrões relacionados ao ElastiCache.

[DAT306: Aprofundamento no Amazon ElastiCache \(re:Invent 2016\)](#)

DAT317: Como o IFTTT usa o ElastiCache for Redis a fim de prever eventos (re:Invent 2016)

O IFTTT é um serviço gratuito que permite às pessoas fazer mais com seus serviços preferidos, desde automatizar tarefas simples até transformar a forma como alguém interage e controla sua casa. O IFTTT usa o ElastiCache for Redis a fim de armazenar o histórico de execução de transações e as previsões de programação, bem como índices de documentos de log no Amazon S3. Assista a esta sessão para saber como o poder de scripts do Lua e os tipos de dados do Redis permitiram que as pessoas realizassem algo que não poderiam ter feito em nenhum outro lugar.

[DAT317: Como o IFTTT usa o ElastiCache for Redis a fim de prever eventos \(re:Invent 2016\)](#)

DAT407: Aprofundamento no Amazon ElastiCache (re:Invent 2015)

Olhe nos bastidores para saber mais sobre o design e a arquitetura da Amazon ElastiCache. Veja os padrões de design comuns das nossas ofertas Memcached e Redis e como os clientes os usaram para operações na memória e obtiveram níveis aprimorados de latência e throughput para

aplicativos. Durante esta sessão, analisamos as melhores práticas, os padrões de design e os antipadrões relacionados ao Amazon ElastiCache.

[DAT407: Aprofundamento no Amazon ElastiCache \(re:Invent 2015\)](#)

SDD402: Aprofundamento no Amazon ElastiCache (re:Invent 2014)

Neste vídeo, examinamos os casos de uso comuns de armazenamento em cache, os mecanismos Memcached e Redis, os padrões que o ajudam a determinar qual mecanismo é o melhor para suas necessidades, aplicação de hash consistente e muito mais como de compilar aplicativos rápidos e dimensionáveis. Frank Wiebe, o principal cientista da Adobe, detalha como a Adobe usa o Amazon ElastiCache para melhorar a experiência dos clientes e escalonar seus negócios.

[DAT402: Aprofundamento no Amazon ElastiCache \(re:Invent 2014\)](#)

DAT307: Aprofundamento na arquitetura e nos padrões de design do Amazon ElastiCache (re:Invent 2013).

Neste vídeo, examinamos o armazenamento em cache, as estratégias de cache, a expansão e o monitoramento. Comparamos também os mecanismos Memcached e Redis. Durante esta sessão, também revisamos as melhores práticas e os padrões de design relacionados ao Amazon ElastiCache.

[DAT307: Aprofundamento na arquitetura e nos padrões de design do Amazon ElastiCache \(AWS re:Invent 2013\)](#)

O que faço agora?

Agora que você tentou o exercício de Conceitos básicos, pode explorar as seções a seguir para saber mais sobre o ElastiCache e as ferramentas disponíveis:

- [Conceitos básicos da AWS](#)
- [Ferramentas para a Amazon Web Services](#)
- [AWS Interface de linha de comando](#)
- [Referência da API do Amazon ElastiCache](#)

Depois de concluir o exercício de Conceitos básicos, você pode ler essas seções para saber mais sobre a administração do ElastiCache:

- [Escolha do tamanho do nó](#)

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

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

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

Gerenciamento de nós

Um nó é o menor bloco de construção de uma implantação do Amazon ElastiCache. Trata-se de um fragmento de tamanho fixo da RAM em armazenamento de rede anexado. Cada nó executa o mecanismo que foi escolhido quando o cluster ou grupo de replicação foi criado ou modificado pela última vez. Cada nó possui seu próprio nome DNS (Serviço de Nomes de Domínio) e porta. Há suporte para vários tipos de nós do ElastiCache, cada um com quantidades diversificadas de memória associada e a potência computacional.

De modo geral, devido ao suporte para fragmentação, as implementações do Redis (modo cluster habilitado) terão vários nós menores. Por outro lado, as implantações do Redis (modo cluster desabilitado) 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](#)
- [Nós e fragmentos do Redis](#)
- [Conexão a nós](#)
- [Tipos de nó compatíveis](#)
- [Reinicialização de nós \(somente para o modo cluster desabilitado\)](#)
- [Substituição de nós](#)
- [Nós reservados do ElastiCache](#)
- [Migração de nós da geração anterior](#)

Algumas operações importantes que envolvem nós são as seguintes:

- [Adição de nós a um cluster](#)
- [Remoção de nós de um cluster](#)

- [Dimensionamento ElastiCache para Redis](#)
- [Localização de endpoints de conexão](#)

Visualizando o status ElastiCache do nó

Usando o [ElastiCache console](#), você pode acessar rapidamente o status do seu ElastiCache nó. O status de um ElastiCache nó indica a integridade do nó. Você pode usar os procedimentos a seguir para visualizar o status do ElastiCache nó no ElastiCache console da Amazon, no AWS CLI comando ou na operação da API.

Os valores de status possíveis para ElastiCache os nós estão na tabela a seguir. Essa tabela também mostra se você será cobrado pelo ElastiCache nó.

Tipo	Faturado	Descrição
available	Faturado	O ElastiCache nó está saudável e disponível.
creating	Não faturado	O ElastiCache nó está sendo criado. O nó está inacessível enquanto é criado.
deleting	Não faturado	O ElastiCache nó está sendo excluído.
modifying	Faturado	O ElastiCache nó está sendo modificado devido a uma solicitação do cliente para modificar o nó.
updating	Faturado	Um estado de atualização indica que uma ou mais das seguintes afirmações são verdadeiras para o ElastiCache nó da Amazon: <ul style="list-style-type: none">• O ElastiCache nó está sendo corrigido como parte

Tipo	Faturado	Descrição
		<p>da atualização do serviço. Para obter mais informações sobre as atualizações do serviço, consulte a página de ajuda do Amazon ElastiCache Managed Maintenance and Service Updates.</p> <ul style="list-style-type: none">• Os grupos de segurança da VPC estão sendo atualizados para o ElastiCache cluster.• O ElastiCache cluster está sendo ampliado ou reduzido.• As configurações de entrega de registros estão sendo modificadas para o ElastiCache cluster.• Uma operação de exclusão do ElastiCache nó está pendente.• A ElastiCache senha do Redis está sendo atualizada/rotacionada usando AWS Secrets Manager

Tipo	Faturado	Descrição
<code>rebooting cache cluster nodes</code>	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.
<code>incompatible_parameters</code>	Não faturado	A Amazon não ElastiCache pode iniciar o nó porque os parâmetros especificados no grupo de parâmetros do nó não são compatíveis com o nó. Reverta as alterações de parâmetro ou as torne compatíveis com o nó para retomar o acesso ao nó. Para obter mais informações sobre os parâmetros incompatíveis, consulte a lista de eventos do ElastiCache nó.

Tipo	Faturado	Descrição
<code>incompatible_network</code>	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">• Não há endereços IP disponíveis na sub-rede em que o ElastiCache nó foi lançado.• A sub-rede mencionada no grupo de ElastiCache sub-redes não existe mais na Amazon Virtual Private Cloud (Amazon VPC).

Tipo	Faturado	Descrição
restore_failed	Não faturado	<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">• As substituições de nó com falha por causa da capacidade da instância insuficiente. Isso normalmente acontece ao executar nós da geração anterior que são end-of-life. No entanto, isso também pode acontecer com a substituição dos nós da geração atual quando AWS não há capacidade sob demanda suficiente para atender à sua solicitação na zona de disponibilidade especificada. Para obter mais informações sobre como corrigir ou remover esses nós, consulte Migração de nós da geração anterior.• Falha na restauração do instantâneo do RDB especificado.• A AWS conta do ElastiCache cluster foi suspensa.• O nó falhou e não pôde ser recuperado.

Tipo	Faturado	Descrição
snapshotting	Faturado	ElastiCache está criando um instantâneo do nó ElastiCache for Redis.

Visualizando o status do ElastiCache nó com o console

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

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

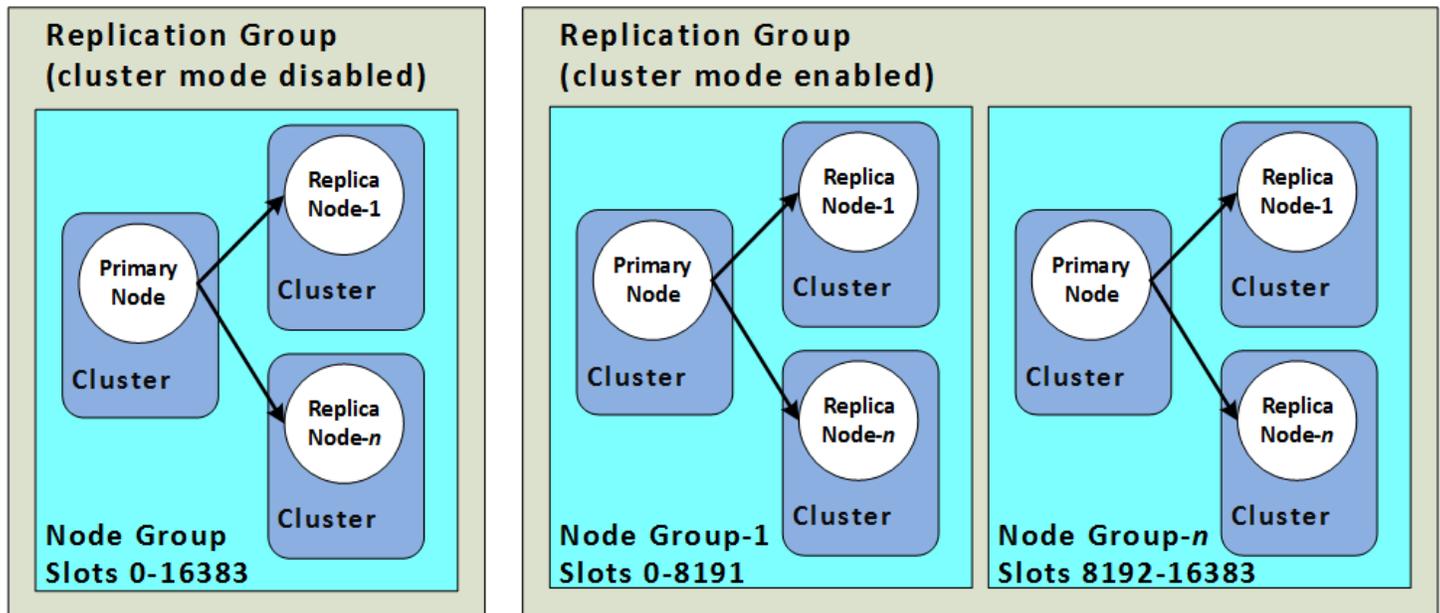
Para visualizar o status do ElastiCache nó usando a ElastiCache API da Amazon, chame `DescribeCacheClusteroperation` with the `ShowCacheNodeInfo` flag para recuperar informações sobre os nós de cache individuais.

Nós e fragmentos do Redis

Um estilhaço (na API e na CLI, um grupo de nós) é uma disposição hierárquica de nós, cada um envolvido em um cluster. Estilhaços oferecem suporte para replicação. Dentro de um estilhaço, um nó funciona como o nó primário de leitura/gravação. Todos os outros nós em um estilhaço funcionam como réplicas somente leitura do nó primário. O Redis versão 3.2 e posterior oferece suporte para

vários estilhaços em um cluster (na API e na CLI, um grupo de replicação). Esse suporte permite particionar os dados em um cluster do Redis (modo cluster habilitado).

O diagrama a seguir ilustra as diferenças entre um cluster Redis (modo cluster desabilitado) e um cluster Redis (modo cluster habilitado).



Os clusters do Redis (modo de cluster habilitado) oferecem suporte para a replicação por meio de fragmentos. A operação de API, [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) lista os grupos de nós com nós membros, a função do nó dentro do grupo de nós, bem como outras informações.

Ao criar um cluster do Redis, você especifica se deseja criar um cluster com o recurso de criação de clusters habilitado. Os clusters do Redis (modo cluster desabilitado) nunca têm mais de um fragmento que pode ser escalado horizontalmente com a adição (até um total de cinco) ou a exclusão de nós de réplica de leitura. Para mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#), [Adicionando uma réplica de leitura, para grupos de replicação do Redis \(Modo cluster desabilitado\)](#) ou [Exclusão de uma réplica de leitura, para grupos de replicação do Redis \(modo cluster desabilitado\)](#). Clusters do Redis (modo cluster desabilitado) também podem ser escalados verticalmente por meio da mudança dos tipos de nó. Para obter mais informações, consulte [Escalabilidade de clusters do Redis \(modo cluster desativado\) com nós de réplica](#).

O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do 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 as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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

Depois que um cluster do Redis (modo cluster habilitado) é criado, ele pode ser alterado (aumento ou redução de escala na horizontal). Para obter mais informações, consulte [Dimensionamento ElastiCache para Redis](#) e [Substituição de nós](#).

Ao criar um novo cluster, você pode preenchê-lo com dados do cluster antigo para que ele não fique vazio. Essa abordagem só funcionará se o grupo de clusters tiver o mesmo número de estilhaços do cluster antigo. Isso pode ser útil se você precisa alterar o tipo de nó ou a versão do mecanismo. Para obter mais informações, consulte [Realização de backups manuais](#) e [Restauração de um backup para um novo cache](#).

Conexão a nós

Antes de tentar se conectar aos nós no seu cluster Redis, você deve ter os endpoints para os nós. Para localizar os endpoints, consulte o seguinte:

- [Localização de endpoints de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para um cluster do Redis \(modo cluster habilitado\) \(console\)](#)
- [Localizar endpoints \(AWS CLI\)](#)
- [Localização de endpoints \(API do ElastiCache\)](#)

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

Note

Para obter mais informações sobre o Redis e os comandos disponíveis do Redis, consulte a página da Web <http://redis.io/commands>.

Para se conectar a um cluster Redis usando a `redis-cli`

1. Conexão à sua instância do Amazon EC2 com o utilitário de conexão de sua escolha.

Note

Para obter instruções sobre como se conectar a uma instância do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon EC2](#).

2. Para criar `redis-cli`, baixe e instale o GNU Compiler Collection (`gcc`). No prompt de comando da sua instância do EC2, digite o seguinte comando e digite `y` no prompt de confirmação.

```
sudo yum install gcc
```

É exibida uma saída semelhante à seguinte.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
```

```
--> Running transaction check

...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB    00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm             | 4.8 MB    00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm               | 2.8 kB    00:00

...(output omitted)...

Complete!
```

3. Baixe e compile o utilitário redis-cli. Esse utilitário está incluído na distribuição do software Redis. No prompt de comando da sua instância do EC2, digite os seguintes comandos:

 Note

Para sistemas Ubuntu, antes de executar o make, execute `make distclean`.

```
wget http://download.redis.io/redis-stable.tar.gz
tar xvzf redis-stable.tar.gz
cd redis-stable
make distclean      # ubuntu systems only
make
```

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

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

É exibido um prompt de comando do Redis semelhante ao seguinte.

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

5. Teste a conexão executando comandos Redis.

Agora, você está conectado ao cluster e pode executar comandos Redis. Os seguintes são alguns comandos de exemplo com suas respostas Redis.

```
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 redis-cli
```

Para conectar-se a nós ou clusters com criptografia Secure Sockets Layer (SSL) (em trânsito habilitado), consulte [Criptografia em trânsito \(TLS\) do ElastiCache](#).

Tipos de nó compatíveis

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das gerações anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

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

Geração atual

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 do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla ação de E/S aprimorada (Redis 7.0.4+)	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m7g.large	6.2	N	N	N	0.937	12,5
cache.m7g.xlarge	6.2	Y	Y	Y	1.876	12,5

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Multiplicação de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha base (Gbps)	Expansão da largura de banda (Gbps)
cache.m7g .2xlarge	6.2	Y	Y	Y	3,75	15
cache.m7g .4xlarge	6.2	Y	Y	Y	7,5	15
cache.m7g .8xlarge	6.2	Y	Y	Y	15	N/D
cache.m7g .12xlarge	6.2	Y	Y	Y	22.5	N/D
cache.m7g .16xlarge	6.2	Y	Y	Y	30	N/D
cache.m6g.large	5.0.6	N	N	N	0.75	10.0
cache.m6g .xlarge	5.0.6	Y	Y	Y	1.25	10.0
cache.m6g .2xlarge	5.0.6	Y	Y	Y	2,5	10.0
cache.m6g .4xlarge	5.0.6	Y	Y	Y	5,0	10.0
cache.m6g .8xlarge	5.0.6	Y	Y	Y	12	N/D

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla conexão de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha base (Gbps)	Expansão da largura de banda (Gbps)
cache.m6g.12xlarge	5.0.6	Y	Y	Y	20	N/D
cache.m6g.16xlarge	5.0.6	Y	Y	Y	25	N/D
cache.m5.large	3.2.4	N	N	N	0.75	10.0
cache.m5.xlarge	3.2.4	Y	N	N	1.25	10.0
cache.m5.2xlarge	3.2.4	Y	Y	Y	2,5	10.0
cache.m5.4xlarge	3.2.4	Y	Y	Y	5,0	10.0
cache.m5.12xlarge	3.2.4	Y	Y	Y	12	N/D
cache.m5.24xlarge	3.2.4	Y	Y	Y	25	N/D
cache.m4.large	3.2.4	N	N	N	0.45	1.2
cache.m4.xlarge	3.2.4	Y	N	N	0.75	2.8
cache.m4.2xlarge	3.2.4	Y	Y	Y	1,0	10.0

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+)	Largura de banda base (Gbps)	Expansão da largura de banda (Gbps)
cache.m4.4xlarge	3.2.4	Y	Y	Y	2.0	10.0
cache.m4.10xlarge	3.2.4	Y	Y	Y	5,0	10.0
cache.t4g.micro	3.2.4	N	N	N	0.064	5,0
cache.t4g.small	5.0.6	N	N	N	0.128	5,0
cache.t4g.medium	5.0.6	N	N	N	0.256	5,0
cache.t3.micro	3.2.4	N	N	N	0.064	5,0
cache.t3.small	3.2.4	N	N	N	0.128	5,0
cache.t3.medium	3.2.4	N	N	N	0.256	5,0
cache.t2.micro	3.2.4	N	N	N	0.064	1.024
cache.t2.small	3.2.4	N	N	N	0.128	1.024
cache.t2.medium	3.2.4	N	N	N	0.256	1.024

Otimizado para memória

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla conexão de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha 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	Y	Y	Y	1,876	12,5
cache.r7g.2xlarge	6.2	Y	Y	Y	3,75	15
cache.r7g.4xlarge	6.2	Y	Y	Y	7,5	15
cache.r7g.8xlarge	6.2	Y	Y	Y	15	N/D
cache.r7g.12xlarge	6.2	Y	Y	Y	22,5	N/D
cache.r7g.16xlarge	6.2	Y	Y	Y	30	N/D
cache.r6g.large	5.0.6	N	N	N	0,75	10,0
cache.r6g.xlarge	5.0.6	Y	Y	Y	1,25	10,0
cache.r6g.2xlarge	5.0.6	Y	Y	Y	2,5	10,0
cache.r6g.4xlarge	5.0.6	Y	Y	Y	5,0	10,0

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla linha de E/S aprimorada (Redis 7.0.4+)	Largura de banda base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6g.8xlarge	5.0.6	Y	Y	Y	12	N/D
cache.r6g.12xlarge	5.0.6	Y	Y	Y	20	N/D
cache.r6g.16xlarge	5.0.6	Y	Y	Y	25	N/D
cache.r5.large	3.2.4	N	N	N	0.75	10.0
cache.r5.xlarge	3.2.4	Y	N	N	1.25	10.0
cache.r5.2xlarge	3.2.4	Y	Y	Y	2,5	10.0
cache.r5.4xlarge	3.2.4	Y	Y	Y	5,0	10.0
cache.r5.12xlarge	3.2.4	Y	Y	Y	12	N/D
cache.r5.24xlarge	3.2.4	Y	Y	Y	25	N/D
cache.r4.large	3.2.4	N	N	N	0.75	10.0
cache.r4.xlarge	3.2.4	Y	N	N	1.25	10.0
cache.r4.2xlarge	3.2.4	Y	Y	Y	2,5	10.0
cache.r4.4xlarge	3.2.4	Y	Y	Y	5,0	10.0

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla conexão de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha base (Gbps)	Expansão da largura de banda (Gbps)
cache.r4.8xlarge	3.2.4	Y	Y	Y	12	N/D
cache.r4.16xlarge	3.2.4	Y	Y	Y	25	N/D

Otimizada para memória com classificação de dados em níveis

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla conexão de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6gd.xlarge	6.2.0	Y	N	N	1.25	10
cache.r6gd.2xlarge	6.2.0	Y	Y	Y	2,5	10
cache.r6gd.4xlarge	6.2.0	Y	Y	Y	5,0	10
cache.r6gd.8xlarge	6.2.0	Y	Y	Y	12	N/D

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Multiplicação de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha base (Gbps)	Expansão da largura de banda (Gbps)
cache.r6gd.12xlarge	6.2.0	Y	Y	Y	20	N/D
cache.r6gd.16xlarge	6.2.0	Y	Y	Y	25	N/D

Otimizadas para rede

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Multiplicação de E/S aprimorada (Redis 7.0.4+)	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	Y	Y	Y	12,5	40
cache.c7gn.2xlarge	6.2	Y	Y	Y	25	50
cache.c7gn.4xlarge	6.2	Y	Y	Y	50	N/D

Tipo de instância	Versão mínima do Redis compatível	E/S aprimorada (Redis 5.0.6+)	Descarregamento de TLS (Redis 6.2.5+)	Múltipla ação de E/S aprimorada (Redis 7.0.4+)	Largura de banda de linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.c7gn.8xlarge	6.2	Y	Y	Y	100	N/D
cache.c7gn.12xlarge	6.2	Y	Y	Y	150	N/D
cache.c7gn.16xlarge	6.2	Y	Y	Y	200	N/D

Tipos de nós compatíveis por região da AWS

Os tipos de nós compatíveis podem variar entre as regiões da AWS. Para obter mais detalhes, consulte [Definição de preço do Amazon ElastiCache](#).

Instâncias de desempenho com capacidade de intermitência

Você pode iniciar os nós de cache T4g, T3-Standard e T2-Standard de uso geral com capacidade de intermitência no Amazon ElastiCache. Esses nós fornecem um nível de linha de base de desempenho de CPU, além da capacidade de intermitência do uso de CPU a qualquer momento até o consumo dos créditos acumulados. Um crédito de CPU oferece o desempenho de um núcleo de CPU completo por um minuto.

Os nós T4g, T3 e T2 do Amazon ElastiCache são configurados como padrão e adequados para workloads com uma utilização média de CPU que seja consistentemente inferior à performance de linha de base da instância. Para intermitências acima da linha de base, o nó gasta os créditos acumulados no seu saldo de créditos de CPU. Se o nó estiver em execução abaixo dos créditos acumulados, o desempenho será gradualmente reduzido para o nível de desempenho de linha de base. Essa redução gradual garante que o nó não experimente uma queda acentuada de desempenho quando seu saldo de créditos acumulados de CPU for esgotado. Para obter mais

informações, consulte [Créditos de CPU e desempenho de linha de base para instâncias de desempenho com capacidade de intermitência](#) no Guia do usuário do Amazon EC2.

A tabela a seguir lista os tipos de nó de desempenho com capacidade de intermitência, a taxa na qual os créditos de CPU são obtidos por hora. Ela também mostra o número máximo de créditos de CPU obtidos que um nó pode acumular e o número de vCPUs por nó. Além disso, fornece o nível de desempenho de linha de base como uma porcentagem do desempenho total de um núcleo (usando uma única vCPU).

Créditos de CPU ganhos por hora	Máximo de créditos obtidos que podem ser acumulados*	vCPUs	Desempenho de linha de base por vCPU	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
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

Créditos de CPU ganhos por hora	Máximo de créditos obtidos que podem ser acumulados*	vCPUs	Desempenho de linha de base por vCPU	Memória (GiB)	Performance de rede
12	288	1	20%	1.55	Baixo a moderado
24	576	2	20%	3.22	Baixo a moderado

* O número de créditos que podem ser acumulados é equivalente ao número de créditos que podem ser obtidos em um período de 24 horas.

** O desempenho de linha de base na tabela é por vCPU. Alguns tamanhos de nós que têm mais de uma vCPU. Para esses, calcule a utilização de CPU da linha de base para o nó multiplicando a porcentagem de vCPU pelo número de vCPUs.

As seguintes métricas de crédito de CPU estão disponíveis para instâncias expansíveis T3 e T4g:

 Note

Essas métricas não estão disponíveis para instâncias expansíveis T2.

- CPUCreditUsage
- CPUCreditBalance

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

Além disso, saiba estes detalhes:

- Todos os tipos de nós da geração atual são criados em uma Virtual Private Cloud (VPC) com base na Amazon VPC por padrão.

- Os arquivos somente de acréscimo (AOF) do Redis não são compatíveis com instâncias T2. As variáveis de configuração do Redis `appendonly` e `appendfsync` não são compatíveis com o Redis versão 2.8.22 e posterior.

Informações relacionadas

- [Recursos e detalhes do produto Amazon ElastiCache](#)
- [Parâmetros específicos do Redis](#)
- [Criptografia em trânsito \(TLS\)](#)

Reinicialização de nós (somente para o modo cluster desabilitado)

Algumas alterações exigem que os nós do cluster sejam reinicializados para serem aplicadas. Por exemplo, para alguns parâmetros, a alteração do valor do parâmetro em um parameter group é aplicada somente após uma reinicialização.

Para clusters do Redis (modo cluster desabilitado), esses parâmetros são:

- `activeresharding`
- banco de dados

Você pode reinicializar um nó usando somente o console do ElastiCache. Você só pode reinicializar um único nó de cada vez. Para reinicializar vários nós, você deve repetir o processo para cada nó.

Alterações de parâmetro do Redis (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster Redis (modo cluster habilitado), siga as etapas a seguir.

- `activeresharding`
- banco de dados

1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
2. Exclua o cluster do Redis (modo cluster habilitado). Consulte [Excluir um cluster](#).
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.

Usar a AWS Management Console

Você pode reinicializar um nó usando o console do ElastiCache.

Para reinicializar um nó (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS que se aplica.
3. No painel de navegação esquerdo, escolha Redis.

É exibida uma lista de clusters que executam o Redis.

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.

Substituição de nós

O Amazon ElastiCache for Redis atualiza frequentemente sua frota com patches e atualizações aplicados às instâncias de forma contínua. No entanto, de tempos em tempos, precisamos executar novamente seus nós do ElastiCache for Redis para aplicar atualizações obrigatórias do sistema operacional ao host subjacente. Essas substituições são necessárias para aplicar atualizações que fortalecem a segurança, a confiabilidade e o desempenho operacional.

Você tem a opção de gerenciar essas substituições a qualquer momento antes da janela agendada para a substituição do nó. Ao gerenciar uma substituição sozinho, sua instância recebe a atualização do sistema operacional quando você executa novamente o nó e a substituição de nó agendada é cancelada. Você pode continuar recebendo alertas que indicam que a substituição do nó ocorrerá. Caso já tenha atenuado manualmente a necessidade da manutenção, você pode ignorar esses alertas.

Note

Os nós de cache de substituição gerados automaticamente pelo Amazon ElastiCache podem ter endereços IP diferentes. Você é responsável por revisar a configuração do aplicativo para garantir que os nós de cache estejam associados aos endereços IP apropriados.

A lista a seguir identifica as ações que você pode realizar quando o ElastiCache programa um de seus nós Redis para substituição. Para agilizar a busca das informações de que você precisa para sua situação, escolha no menu a seguir.

- [Do nothing](#): deixar que o Amazon ElastiCache substitua o nó conforme programado.
- [Change your maintenance window](#): alterar sua janela de manutenção para um melhor momento.
- Configurações do Redis (modo cluster habilitado)
 - [Replace the only node in any Redis cluster](#): um procedimento para substituir um nó em um cluster do Redis usando backup e restauração.
 - [Replace a replica node in any Redis cluster](#): um procedimento para substituir uma réplica de leitura em qualquer cluster do Redis aumentando e diminuindo-se a contagem de réplicas sem tempo de inatividade do cluster.
 - [Replace any node in a Redis \(cluster mode enabled\) shard](#): um procedimento dinâmico sem tempo de inatividade do cluster para substituir um nó em um cluster do Redis (modo cluster habilitado) com aumento ou redução de escala na horizontal.

- Configurações do Redis (modo cluster desabilitado)
 - [Replace the only node in any Redis cluster](#): procedimento para substituir qualquer nó em um cluster do Redis usando backup e restauração.
 - [Replace a replica node in any Redis cluster](#): um procedimento para substituir uma réplica de leitura em qualquer cluster do Redis aumentando e diminuindo-se a contagem de réplicas sem tempo de inatividade do cluster.
 - [Replace a node in a Redis \(cluster mode disabled\) cluster](#): procedimento para substituir um nó em um cluster do Redis (modo cluster desabilitado) usando replicação.
 - [Replace a Redis \(cluster mode disabled\) read-replica](#): um procedimento para substituir manualmente uma réplica de leitura em um grupo de replicação do Redis (modo cluster desabilitado).
 - [Replace a Redis \(cluster mode disabled\) primary node](#): um procedimento para substituir manualmente o nó primário em um grupo de replicação Redis (modo cluster desabilitado).

Opções de substituição de nós Redis

- Não fazer nada: se você não fizer nada, o ElastiCache substituirá o nó conforme programado.

Para as configurações sem cluster com o failover automático habilitado, os clusters no Redis 5.0.6 e posterior concluem a substituição enquanto o cluster permanece on-line e atende às solicitações de gravação recebidas. Para os clusters habilitados para failover no Redis 54.0.10 ou abaixo, você poderá observar uma breve interrupção da gravação, de até alguns segundos, associada à atualização do DNS.

Se o nó for membro de um cluster habilitado para failover automático, o ElastiCache for Redis fornecerá maior disponibilidade durante a aplicação de patches, as atualizações e outras substituições de nós relacionadas à manutenção.

As configurações de cluster ElastiCache for Redis que estão definidas para usar clientes de cluster ElastiCache for Redis agora são concluídas enquanto o cluster atende às solicitações de gravação recebidas.

Para as configurações sem cluster com o failover automático habilitado, os clusters no Redis 5.0.6 e posterior concluem a substituição enquanto o cluster permanece on-line e atende às solicitações de gravação recebidas. Para os clusters habilitados para failover no Redis 54.0.10 ou abaixo, você poderá observar uma breve interrupção da gravação, de até alguns segundos, associada à atualização do DNS.

Se o nó for autônomo, o Amazon ElastiCache primeiro ativará um nó de substituição e depois fará a sincronização a partir do nó existente. O nó existente não está disponível para solicitações de serviço durante esse período. Uma vez concluída a sincronização, o nó existente é encerrado, e o novo nó assume seu lugar. O ElastiCache faz o melhor esforço possível para reter seus dados durante esta operação.

- Mudar sua janela de manutenção: para eventos de manutenção programada, você recebe um email ou uma notificação do evento do ElastiCache. Nesses casos, se você mudar sua janela de manutenção antes da hora de substituição programada, o nó será substituído no novo horário. Para obter mais informações, consulte as informações a seguir.
 - [Modificando um cluster ElastiCache](#)
 - [Modificação de um grupo de replicação](#)

Note

A capacidade de alterar sua janela de substituição movendo sua janela de manutenção só está disponível quando a notificação do ElastiCache inclui uma janela de manutenção. Se a notificação não inclui uma janela de manutenção, não é possível alterar a janela de substituição.

Por exemplo, digamos que seja quinta-feira, 9 de novembro, às 15h e a próxima janela de manutenção seja sexta-feira, 10 de novembro, às 17h. Veja estes três cenários e seus resultados:

- Você altera sua janela de manutenção para sexta-feira, 16h (após a data e hora atual e antes da próxima janela de manutenção programada). O nó é substituído na sexta-feira, 10 de novembro, às 16h.

- Você altera sua janela de manutenção para sábado, 16h (após a data e hora atual e a próxima janela de manutenção programada). O nó é substituído no sábado, 11 de novembro, às 16h.
- Você altera sua janela de manutenção para quarta-feira, 16h (dia da semana anterior à data e hora atual). O nó é substituído na próxima quarta-feira, 15 de novembro, às 16h.

Para obter instruções, consulte [Gerenciamento da manutenção](#).

- Substituir o único nó em qualquer cluster do Redis: caso o cluster não tenha réplicas de leitura, você pode usar o procedimento a seguir para substituir o nó.

Para substituir o único nó usando backup e restauração

1. Crie um snapshot do cluster do nó. Para 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 [Excluir um cluster](#).
 4. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
- Substituir um nó de réplica em qualquer cluster do Redis: para substituir um cluster de réplica, aumente sua contagem de réplicas. Para fazer isso, adicione uma réplica e diminua a contagem de réplicas removendo a réplica que deseja substituir. Esse processo é dinâmico e não tem tempo de inatividade do cluster.

 Note

Caso o fragmento ou o grupo de replicação já tenha cinco réplicas, reverta as etapas 1 e 2.

Para substituir uma réplica em qualquer cluster Redis

1. Aumente a contagem de réplicas adicionando uma réplica ao estilhaço ou ao grupo de replicação. Para obter mais informações, consulte [Aumento do número de réplicas em um fragmento](#).

2. Exclua a réplica que você deseja substituir. Para obter mais informações, consulte [Diminuição do número de réplicas em um fragmento](#).
 3. Atualize os endpoints no aplicativo.
- Substituir qualquer nó em um fragmento do Redis (modo cluster habilitado): para substituir o nó em um cluster sem nenhum tempo de inatividade, use a refragmentação online. Primeiro adicione um estilhaço por expansão e, depois, exclua o estilhaço com o nó a ser substituído por redução.

Para substituir qualquer nó em um cluster do Redis (modo cluster habilitado)

1. Aumentar: adicione mais um estilhaço com a mesma configuração que o estilhaço existente que contém o nó a ser substituído. Para obter mais informações, consulte [Adição de fragmentos com refragmentação online](#).
 2. Diminuir: exclua o estilhaço com o nó a ser substituído. Para obter mais informações, consulte [Remoção de fragmentos com refragmentação online](#).
 3. Atualize os endpoints no aplicativo.
- Substituir um nó em um cluster do Redis (modo cluster desabilitado): se o cluster for um cluster do Redis (modo cluster desabilitado) sem nenhuma réplica de leitura, use o procedimento a seguir para substituir o nó.

Como substituir o nó usando a replicação (somente para o modo de cluster desabilitado)

1. Adicione a replicação ao cluster com o nó programado para substituição como primário. Não habilite o recurso Multi-AZ nesse cluster. Para obter instruções, consulte [Para adicionar replicação a um cluster Redis sem estilhaços](#).
2. Adicione uma réplica de leitura ao cluster. Para obter instruções, consulte [Para adicionar nós a um cluster \(console\)](#).
3. Promova a réplica de leitura recém-criada para primário. Para obter instruções, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Redis \(modo cluster desabilitado\)](#).
4. Exclua o nó agendado para substituição. Para obter instruções, consulte [Remoção de nós de um cluster](#).

5. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
- Substituir uma réplica de leitura do Redis (modo cluster desabilitado): se o nó for uma réplica de leitura, substitua o nó.

Se o cluster tiver apenas um nó de réplica e o recurso Multi-AZ estiver habilitado, você deverá desabilitar o Multi-AZ antes de excluir a réplica. Para obter instruções, consulte [Modificação de um grupo de replicação](#).

Para substituir uma réplica de leitura do Redis (modo cluster desabilitado)

1. Exclua a réplica programada para substituição. Para obter instruções, consulte o seguinte:
 - [Diminuição do número de réplicas em um fragmento](#)
 - [Remoção de nós de um 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 o seguinte:
 - [Aumento do número de réplicas em um fragmento](#)
 - [Adicionando uma réplica de leitura, para grupos de replicação do Redis \(Modo cluster desabilitado\)](#)
 3. No seu aplicativo, substitua o endpoint da réplica antiga pelo endpoint da nova réplica.
 4. Se você tiver desabilitado o recurso Multi-AZ no início, reabilite-o agora. Para obter instruções, consulte [Habilitar Multi-AZ](#).
- Substituir um nó primário do Redis (modo cluster desabilitado): se o nó for o nó primário, primeiro promova uma réplica de leitura para primário. Depois, exclua a réplica usada para ser o nó principal.

Se o cluster tiver apenas uma réplica e o recurso Multi-AZ estiver habilitado, você deverá desabilitar o Multi-AZ antes de excluir a réplica na etapa 2. Para obter instruções, consulte [Modificação de um grupo de replicação](#).

Para substituir um nó primário do Redis (modo cluster desabilitado)

1. Promover uma réplica de leitura para primário. Para obter instruções, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Redis \(modo cluster desabilitado\)](#).
2. Exclua o nó programado para substituição (o primário antigo). Para obter instruções, consulte [Remoção de nós de um 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 [Adicionando uma réplica de leitura, para grupos de replicação do Redis \(Modo cluster desabilitado\)](#).

4. No seu aplicativo, substitua o endpoint do nó antigo pelo endpoint do novo nó.
5. Se você tiver desabilitado o recurso Multi-AZ no início, reabilite-o agora. Para obter instruções, consulte [Habilitar Multi-AZ](#).

Nós reservados do ElastiCache

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.

Para ver se os nós reservados são uma economia para seus casos de uso, primeiro determine o tamanho do nó e o número de nós necessários. Depois, estime o uso do nó e compare o custo total de usar nós sob demanda e nós reservados. Você pode combinar e misturar o uso de nós reservados e sob demanda nos seus clusters. Para obter informações de preços, consulte [Definição de preço do Amazon ElastiCache](#).

Note

Os nós reservados não são flexíveis; eles se aplicam apenas ao tipo exato de instância reservada.

Gerenciamento de custos com nós reservados

Reservar um ou mais nós pode ser uma maneira de reduzir custos. Para nós reservados, é necessário pagar uma taxa adiantada que depende do tipo de nó e da duração da reserva, um ou três anos. Essa cobrança é muito menor do que a cobrança de uso por hora dos nós sob demanda.

Para ver se os nós reservados são uma economia para seus casos de uso, primeiro determine o tamanho do nó e o número de nós necessários. Depois, estime o uso do nó e compare o custo total de usar nós sob demanda e nós reservados. Você pode combinar e misturar o uso de nós reservados e sob demanda nos seus clusters. Para obter informações de preços, consulte [Definição de preço do Amazon ElastiCache](#).

A região da AWS, o tipo de nó e o tempo de duração devem ser escolhidos no ato da compra e não podem ser alterados posteriormente.

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

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

Tópicos

- [Ofertas de nós reservados padrão](#)
- [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](#)

Ofertas de nós reservados padrão

Ao comprar uma instância de nó reservado padrão (RI) no Amazon ElastiCache, você adquire um compromisso de obter uma taxa com desconto sobre um tipo de instância de nó e região da AWS específicos para a duração da instância de nó reservado. Para usar uma instância do nó reservada do Amazon ElastiCache, uma nova instância do nó do ElastiCache deve ser criada, como faria para uma instância sob demanda.

A nova instância do nó que você criar deve corresponder exatamente às especificações da instância do nó reservada. Se as especificações da nova instância do nó corresponderem às de uma instância do nó reservada existente para a sua conta, será cobrada a taxa de desconto oferecida para a instância do nó reservada. Caso contrário, uma taxa sob demanda será cobrada para a instância do nó. Esses RIs padrão estão disponíveis a partir das famílias de instâncias R5 e M5.

Note

Todos os três tipos de ofertas discutidas a seguir estão disponíveis para períodos de vigência de um e três anos.

Tipos de oferta

Sem adiantamento O RI fornece acesso a uma instância ElastiCache reservado sem a necessidade de pagamento adiantado. Sua instância do ElastiCache reservado sem adiantamento cobra uma taxa por hora com desconto a cada hora dentro do prazo, independentemente do uso.

Adiantamento parcial A RI exige que uma parte da instância reservada do ElastiCache seja paga antecipadamente. As horas restantes do período de vigência serão cobradas com base em uma taxa horária com desconto, independentemente do uso. Essa opção é a substituição da opção antiga de utilização pesada, que é explicada na próxima seção.

A RI adiantamento integral exige que o pagamento integral seja feito no início do período de vigência da RI. Você não incorrem em outros custos pelo restante do período, independentemente do número de horas usadas.

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, independentemente de você estar ou não usando o nó reservado.	O custo geral mais baixo se você planeja executar seus nós reservados mais de 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	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.

Oferta	Custo inicial	Taxa de uso	Vantagem
		reservado está sendo executado.	
Uso sob demanda (sem nós reservados)	Nenhum	Taxa horária mais alta. Aplicada sempre que o nó está sendo executado.	Maior custo por hora.

Para obter mais informações, consulte [Definição de preço Amazon ElastiCache](#).

Obtenção de informações sobre ofertas de nós reservados

Antes de comprar nós reservados, você pode obter informações sobre ofertas de nós reservados disponíveis.

Os exemplos a seguir mostram como obter a definição de preço e informações sobre ofertas de nós reservados disponíveis usando o AWS Management Console, a AWS CLI e a API do ElastiCache.

Tópicos

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

Obtenção de informações sobre ofertas de nós reservados (console)

Para obter definição de preço e outras informações sobre ofertas de clusters reservados disponíveis usando o 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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, selecione Reserved Nodes.
3. Escolha Purchase Reserved Nodes (Comprar nós reservados).
4. Em Engine (Mecanismo), escolha Redis.
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 informações sobre preços e outras informações sobre ofertas de nós reservados disponíveis, digite o seguinte comando em um prompt de comando:

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

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 31536000,
```

```
    "FixedPrice": X.X,  
    "UsagePrice": X.X,  
    "ProductDescription": "redis",  
    "OfferingType": "No Upfront",  
    "RecurringCharges": [  
      {  
        "RecurringChargeAmount": X.XXX,  
        "RecurringChargeFrequency": "Hourly"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [describe-reserved-cache-nodes-offerings](#) na AWS CLI Referência.

Obtenção de informações sobre ofertas de nós reservados (API do ElastiCache)

Para obter informações sobre preços e informações sobre ofertas de nós reservados disponíveis, chame a ação `DescribeReservedCacheNodesOfferings`.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodesOfferings  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [DescribeReservedCacheNodesOfferings](#) na Referência da API do ElastiCache.

Compra de um nó reservado

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

Important

Seguir os exemplos desta seção gerará cobranças na sua conta da AWS que você não poderá reverter.

Tópicos

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

Compra de um nó reservado (console)

Este exemplo mostra a compra de uma oferta de nó reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, com um ID de nó reservado de myreservationID.

O procedimento a seguir usa o AWS Management Console para comprar a oferta de nó reservado por ID de oferta.

Para comprar nós reservados

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha o link Reserved nodes (Nós reservados).
3. Escolha o botão Purchase reserved nodes (Comprar nós reservados).
4. Em Engine (Mecanismo), escolha Redis.
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-offering](#) na AWS CLI Referência.

Compra de um nó reservado (API do ElastiCache)

O exemplo a seguir mostra a compra de uma oferta de nó reservado específica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, com um ID de cluster reservado de myreservationID.

Chame a operação PurchaseReservedCacheNodesOffering com os seguintes parâmetros:

- ReservedCacheNodesOfferingId = 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
- ReservedCacheNodeID = myreservationID
- CacheNodeCount = 1

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=PurchaseReservedCacheNodesOffering  
  &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  
  &ReservedCacheNodeID=myreservationID  
  &CacheNodeCount=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [PurchaseReservedCacheNodesOffering](#) na Referência da API do ElastiCache.

Obtenção de informações sobre seus nós reservados

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

Tópicos

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

Obtenção de informações sobre seus nós reservados (console)

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

Para obter informações sobre seus nós reservados comprados

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha o link Reserved nodes (Nós reservados).

Os nós reservados para sua conta aparecem na lista Reserved nodes (Nós reservados). Você pode escolher qualquer um dos nós reservados na lista para ver informações detalhadas sobre o nó reservado no painel de detalhes na parte inferior do console.

Obtenção de informações sobre seus nós reservados (AWS CLI)

Para obter informações sobre nós reservados para sua a conta da AWS, digite o seguinte comando em um prompt de comando:

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

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
  "CacheNodeType": "cache.xx.small",
```

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

Para obter mais informações, consulte [describe--reserved-cache-nodes](#) na Referência da AWS CLI.

Obtenção de informações seus nós reservados (API do ElastiCache)

Para obter informações sobre nós reservados para sua conta da AWS, execute a operação `DescribeReservedCacheNodes`.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodes  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte [DescribeReservedCacheNodes](#) na Referência da API do ElastiCache.

Migração de nós da geração anterior

Os nós de geração anterior são tipos de nó que estão sendo eliminados gradualmente. Se você não tiver clusters existentes usando um tipo de nó de geração anterior, o ElastiCache não oferece suporte à criação de novos clusters com esse tipo de nó.

Devido à quantidade limitada de tipos de nó de geração anterior, não podemos garantir uma substituição bem-sucedida quando um nó se torna não íntegro em seu(s) cluster(s). Nesse cenário, a disponibilidade do cluster pode ser afetada negativamente.

Recomendamos que você migre seu(s) cluster(s) para um novo tipo de nó para melhor disponibilidade e performance. Para obter um tipo de nó recomendado para migrar, consulte [Caminhos de atualização](#). Para obter uma lista completa dos tipos de nó com suporte e dos tipos de nó da geração anterior no ElastiCache, consulte [Tipos de nó compatíveis](#).

Migração de nós em um cluster do Redis

O procedimento a seguir descreve como migrar um cluster do Redis usando o console do ElastiCache. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo. Dependendo da configuração do cluster, você pode ver os seguintes tempos de inatividade. As estimativas a seguir podem diferir com base em suas configurações específicas:

- O modo cluster desabilitado (nó único) pode ver aproximadamente 60 segundos, principalmente devido à propagação do DNS.
- O modo de cluster desabilitado (com o nó de réplica) pode ver aproximadamente um segundo para clusters que executam o Redis 5.0.6 e posterior. Todas as versões inferiores podem experimentar aproximadamente 10 segundos.
- O modo cluster habilitado pode ver aproximadamente 1 segundo.

Para modificar um tipo de nó de cluster do Redis usando o console:

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

Use a API [modify-replication-group](#) conforme mostrado a seguir:

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group /  
  --replication-group-id my-replication-group /  
  --cache-node-type new-node-type /  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-replication-group ^  
  --cache-node-type new-node-type ^  
  --apply-immediately
```

Nesse cenário, o valor de *new-node-type* é o tipo de nó para o qual você está migrando.

Passando o parâmetro `--apply-immediately`, a atualização será aplicada imediatamente quando o grupo de replicação fizer a transição do status `modifying` (modificando) para `available` (disponível).

Se `Apply immediately` (Aplicar imediatamente) não for escolhida, o processo de migração será realizado durante a próxima janela de manutenção do cluster.

Note

Se você não conseguir modificar o cluster com um erro `InvalidCacheClusterState`, você precisa remover primeiro um nó com falha de restauração.

Correção ou remoção de nós com falha na restauração

O procedimento a seguir descreve como corrigir ou remover nós com falha na restauração pelo cluster do Redis. Para saber mais sobre como nós do ElastiCache entram em um estado de falha na restauração, consulte [Visualizando o status ElastiCache do nó](#). É recomendável primeiramente remover eventuais nós em um estado de falha na restauração e, em seguida, migrar os nós da geração anterior restantes no cluster do ElastiCache para um tipo de nó de uma geração mais nova e, por fim, readicionar o número de nós necessário.

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

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).
3. Na lista de clusters, escolha o cluster do qual você deseja remover um nó.
4. Na lista de fragmentos, escolha o fragmento do qual você deseja remover um nó. Ignore esta etapa se o modo cluster estiver desabilitado para o cluster.
5. Na lista de nós, escolha o nó com um status de `restore-failed`.
6. Escolha Actions (Ações) e, em seguida, escolha Delete node (Excluir nó).

Depois de remover os nós com falha na restauração do cluster do ElastiCache, você poderá migrar para um tipo de geração mais novo. Para obter mais informações, consulte acima em [Migração de nós em um cluster do Redis](#).

Para readicionar nós ao cluster do ElastiCache, consulte [Adição de nós a um cluster](#).

Gerenciamento de clusters

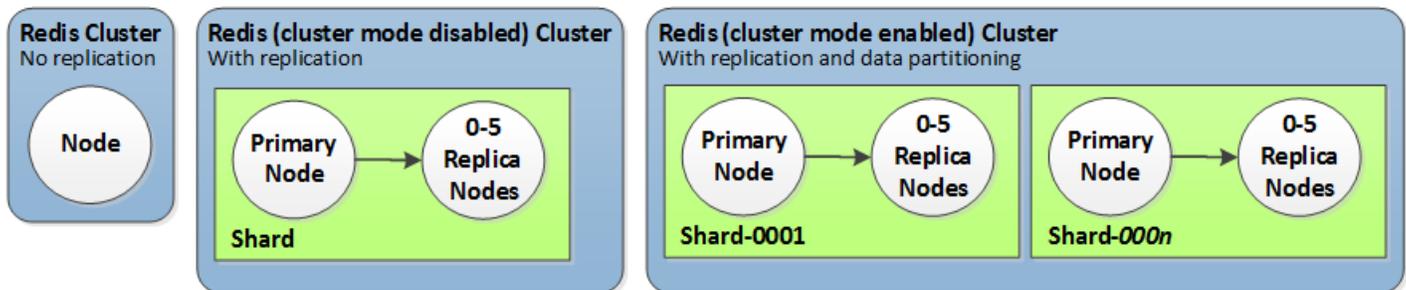
Um cluster é uma coleção de um ou mais nós de cache, todos executando uma instância do software do mecanismo de cache do Redis. Ao criar um cluster, especifique o mecanismo e a versão para que todos os nós usem.

O diagrama a seguir ilustra um cluster típico do Redis. Os clusters do Redis podem conter um nó único ou até seis nós dentro de um fragmento (API/CLI: grupo de nós). Um nó único do cluster do Redis (modo cluster desabilitado) não possui fragmento e um cluster do Redis com vários nós (modo cluster desabilitado) tem um único fragmento. Clusters do Redis (modo cluster habilitado) podem ter até 500 fragmentos, com seus dados particionados nos fragmentos. O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do 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 as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para ter mais informações, consulte [Criação de um grupo de sub-redes](#). Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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

Quando você possui múltiplos nós em um estilhaço, um dos nós é um nó primário de leitura/ gravação. Todos os outros nós no estilhaço são réplicas somente leitura.

Os clusters do Redis normais são semelhantes ao seguinte.



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 a ElastiCache API e AWS CLI os comandos. 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 instância do Amazon EC2. Ao executar seu cluster em uma nuvem privada virtual (VPC) com base no serviço da Amazon VPC, é possível acessá-lo externamente à AWS. Para ter mais informações, consulte [Acesso a recursos do ElastiCache de fora da AWS](#).

Para obter uma lista das versões do Redis compatíveis, consulte [Versões do ElastiCache para Redis com suporte](#).

Escolher um tipo de rede

O ElastiCache oferece suporte às versões 4 e 6 do protocolo Internet (IPv4 e IPv6), permitindo que você configure seu cluster para aceitar:

- somente conexões IPv4,
- somente conexões IPv6,
- conexões IPv4 e IPv6 (pilha dupla)

O IPv6 é compatível com workloads usando o mecanismo Redis versão 6.2 em diante em todas as instâncias criadas no [sistema Nitro](#). Não há custo adicional para acessar o ElastiCache por meio do IPv6.

Note

A migração de clusters criados antes da disponibilidade do IPV6/pilha dupla não tem suporte. A alternância entre tipos de rede em clusters recém-criados também não tem suporte.

Configurar sub-redes para o tipo de rede

Se você criar um cluster em uma Amazon VPC, deverá especificar um grupo de sub-rede. O ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. Os clusters do ElastiCache exigem uma sub-rede de pilha dupla com endereços IPv4 e IPv6 atribuídos a eles para operar no modo de pilha dupla e uma sub-rede somente IPv6 para operar somente como IPv6.

Usar pilha dupla

Ao usar o ElastiCache para Redis no modo de cluster habilitado, do ponto de vista de uma aplicação, 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.

Para manter a compatibilidade com versões anteriores de todos os clientes existentes, é apresentada a descoberta de IP, que permite selecionar o tipo de IP (ou seja, IPv4 ou IPv6) para

anunciar no protocolo de descoberta. Embora isso limite a descoberta automática a apenas um tipo de IP, a pilha dupla ainda é benéfica para workloads habilitadas no modo de cluster, pois permite migrações (ou reversões) de um tipo de IP de descoberta IPv4 para um IPv6 sem tempo de inatividade.

Clusters do ElastiCache de pilha dupla habilitados para TLS

Quando o TLS está habilitado para clusters do ElastiCache, as funções de descoberta de cluster (`cluster slots`, `cluster shards` e `cluster nodes`) retornam nomes de host em vez de IPs. Os nomes de host são então usados em vez de IPs para se conectar ao cluster do ElastiCache e realizar um handshake de TLS. Isso significa que os clientes não serão afetados pelo parâmetro de descoberta de IP. Para clusters habilitados para TLS, o parâmetro de descoberta de IP não tem efeito no protocolo IP preferencial. Em vez disso, o protocolo IP usado será determinado pelo protocolo IP que o cliente prefere ao resolver nomes de host DNS.

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

Usar a AWS Management Console

Ao criar um cluster usando o AWS Management Console, em Connectivity (Conectividade), escolha um tipo de rede, IPv4, IPv6 ou pilha dupla. Se você estiver criando um cluster do Redis (modo de cluster habilitado) e escolher pilha dupla, deverá selecionar um Discovery IP type (Tipo de IP de descoberta): IPv6 ou IPv4.

Para obter mais informações, consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#) ou [Criação de um Redis \(modo cluster desabilitado\) \(console\)](#).

Ao criar um grupo de replicação usando o AWS Management Console, escolha um tipo de rede: IPv4, IPv6 ou pilha dupla. Se você escolher pilha dupla, deverá selecionar um Discovery IP type (Tipo de IP de descoberta): IPv6 ou IPv4.

Para obter mais informações, consulte [Criação de um grupo de replicação do Redis \(modo cluster desabilitado\) do princípio](#) ou [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio](#).

Uso da CLI

Ao criar um cluster de cache usando a CLI, você usa o comando [create-cache-cluster](#) 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 desabilitado usando a CLI, você usa o comando [create-replication-group](#) e especifica os parâmetros NetworkType e IPDiscovery:

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 habilitado e usar IPv4 para descoberta de IP usando a CLI, você usa o comando [create-replication-group](#) e especifica os parâmetros `NetworkType` e `IPDiscovery`:

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv4 \
  --region us-east-1
```

Para Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv4 ^
  --region us-east-1
```

Ao criar um grupo de replicação com o modo de cluster habilitado e usar IPv6 para descoberta de IP usando a CLI, você usa o comando [create-replication-group](#) e especifica os parâmetros `NetworkType` e `IPDiscovery`:

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
```

Classificação de dados em níveis

Os clusters que compõem um grupo de replicação e usam um tipo de nó da família r6gd têm seus dados classificados em níveis entre memória e armazenamento local em Solid State Drives (SSDs – Unidades de estado sólido). A classificação de dados em níveis fornece uma nova opção de performance de preço para workloads do Redis com SSDs de menor custo em cada nó de cluster, além do datastore na memória. Essa modalidade é ideal para workloads que acessam regularmente até 20% do conjunto de dados geral e para aplicações que podem tolerar latência adicional ao acessar dados em SSD.

Em clusters 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 usado menos recentemente (LRU) para mover automaticamente itens acessados com pouca frequência da memória para o SSD. Quando os dados no SSD são acessados posteriormente, eles são movidos de volta para a memória de forma ElastiCache 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 a classificação por níveis, as próprias chaves sempre permanecem na memória, enquanto a LRU controla a colocação de valores na memória versus disco. Em geral, recomendamos que seus tamanhos de chave sejam menores do que seus tamanhos de valor ao usar a classificação por níveis de dados.

A classificação de dados em níveis foi projetada para causar impacto mínimo na performance das workload da aplicação. Por exemplo, presumindo valores de string de 500 bytes, você pode esperar uma média de mais 300 microssegundos de latência para solicitações de dados armazenados em SSD em comparação com solicitações de dados em memória.

Com o maior tamanho de nó de classificação de dados em níveis (cache.r6gd.16xlarge), você pode armazenar até 1 petabyte em um só cluster de 500 nós (500 TB ao usar 1 réplica de leitura). O armazenamento em camadas de dados é compatível com todos os comandos e estruturas de dados do Redis suportados no ElastiCache Para usar esse recurso, não é necessário promover alterações no lado do cliente.

Tópicos

- [Práticas recomendadas](#)
- [Limitações](#)
- [Definição de preço](#)
- [Monitoramento](#)
- [Como usar a classificação de dados em níveis](#)
- [Como restaurar dados do backup para clusters com a classificação de dados em níveis ativada](#)

Práticas recomendadas

Recomendamos seguir estas práticas recomendadas:

- A classificação de dados em níveis é ideal para workloads que acessam regularmente até 20% do conjunto de dados geral e para aplicações que podem tolerar latência adicional ao acessar dados em SSD.
- Ao usar a capacidade SSD disponível em nós em níveis de dados, recomendamos que o tamanho do valor seja maior do que o tamanho da chave. Quando os itens são movidos entre DRAM e SSD, as chaves sempre permanecerão na memória e somente os valores serão movidos para a camada SSD.

Limitações

A classificação de dados em níveis tem as seguintes limitações:

- Você só pode usar a classificação de dados em níveis em clusters que fazem parte de um grupo de replicação.
- O tipo de nó usado deve ser da família r6gd, que está disponível nas seguintes regiões: us-east-2, us-east-1, us-west-2, us-west-1, eu-west-1, eu-central-1, eu-north-1, eu-west-3, ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-south-1, ca-central-1 e sa-east-1.
- O mecanismo deve ser o Redis 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 ter mais informações, consulte [Dimensionamento ElastiCache para Redis](#).
- O ajuste de escala automático é compatível em clusters que usam divisão de dados em camadas para Redis versão 7.0.7 e posteriores. Para obter mais informações, consulte [Auto Scaling ElastiCache para clusters Redis](#).
- A divisão de dados em camadas só são compatíveis com as políticas volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu e noeviction.
- A gravação sem bifurcação é compatível com o Redis versão 7.0.7 e posteriores. Para ter mais informações, consulte [Como a sincronização e o backup são implementados](#).
- Itens maiores que 128 MiB não são movidos para o SSD.

Definição de preço

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

Monitoramento

ElastiCache for Redis oferece métricas projetadas especificamente para monitorar os clusters de desempenho que usam camadas de dados. Para monitorar a proporção de itens na DRAM em comparação com o SSD, é possível usar a métrica de `CurrlItems` em [Métricas para Redis](#). Você pode calcular a porcentagem como: $(\text{CurrlItems com Dimensão: Nível} = \text{Memória} * 100) / (\text{CurrlItems sem filtro de dimensão})$. Quando a porcentagem de itens na memória diminuir para menos de 5 por cento, recomendamos que você considere aumentar a escala horizontalmente para [clusters no Cluster Mode Disabled \(Modo de cluster desabilitado\)](#) ou aumentar a escala verticalmente para [clusters desabilitados para Modo cluster](#). Para mais informações, consulte [Métricas para clusters Redis que usam classificação de dados em níveis em Métricas para Redis](#).

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 [Criar um cluster](#).

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 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 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 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 (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. Cluster mode enabled (scale out) (Modo cluster habilitado, aumento de escala na horizontal): escolha esta opção para um cluster do Redis (modo cluster habilitado).
 3. Node Type (Tipo de nó) – Especifique cache.r6gd.xlarge ou qualquer outro tipo de nó da família r6gd.
 4. Number of Shards – Escolha o número de estilhaços desejados no novo cluster (API/CLI: grupos de nós).
 5. Replicas per Shard – Escolha o número de nós de réplica de leitura desejados em cada estilhaço.
 6. Slots and keyspaces (Slots e espaços de chaves): escolha como deseja que as chaves sejam distribuídas entre os fragmentos. Se você optar por especificar as distribuições de chaves, complete a tabela especificando os intervalos de chaves para cada estilhaço.
 7. Availability zone(s) – especifique como você deseja que as zonas de disponibilidade do cluster sejam selecionadas.

8. Port – Somente altere esse valor se quiser que o novo cluster use uma porta diferente.
 9. Choose a VPC – Escolha a VPC na qual criar esse cluster.
 10. Parameter Group – Escolha um parameter group que reserve memória suficiente para a sobrecarga 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 [Criar um cluster](#).

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 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 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 cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Preparação de um cluster

Veja a seguir instruções sobre como criar um cluster usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

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

- [Determinação dos seus requisitos](#)
- [Escolha do tamanho do nó](#)

Determinação dos seus requisitos

Preparação

Conhecer as respostas às seguintes perguntas ajuda a tornar a criação do cluster mais simples:

- De qual tipo de instância de nó você precisa?

Para obter orientação sobre como escolher um tipo de nó de instância, consulte [Escolha do tamanho do nó](#).

- Você executará seu cluster em uma nuvem privada virtual (VPC) baseada na Amazon VPC?

Important

Se você estiver executando seu cluster em uma VPC, crie um grupo de sub-redes na mesma VPC antes de começar a criar um cluster. Para ter mais informações, consulte [Sub-redes e grupos de sub-redes](#).

ElastiCache foi projetado para ser acessado de dentro AWS usando o Amazon EC2. No entanto, ao iniciá-lo em uma VPC com base na Amazon VPC e o seu cluster em uma VPC, você pode fornecer acesso de fora da AWS. Para ter mais informações, consulte [Acesso a recursos do ElastiCache de fora da AWS](#).

- Você precisa personalizar qualquer valor de parâmetro?

Se você fizer isso, crie um grupo de parâmetro personalizado. Para ter mais informações, consulte [Criar um parameter group](#).

Se você estiver executando o Redis, considere definir `reserved-memory` ou `reserved-memory-percent`. Para ter mais informações, consulte [Gerenciamento da memória reservada](#).

- Você precisa criar o próprio grupo de segurança da VPC?

Para obter mais informações, consulte [Segurança na sua VPC](#).

- Como você pretende implementar a tolerância a falhas?

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

Tópicos

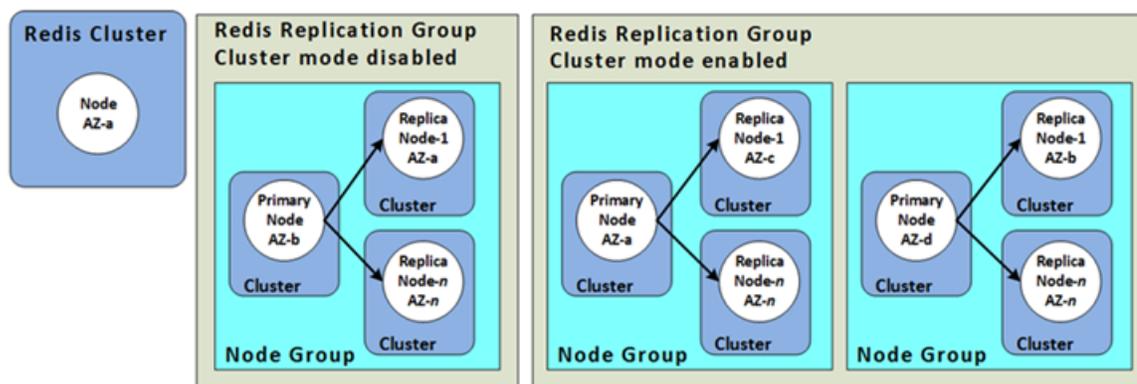
- [Requisitos de memória e processador](#)
- [Configuração do cluster do Redis](#)
- [Requisitos de escalabilidade](#)
- [Requisitos de acesso](#)
- [Requisitos de região, zona de disponibilidade e zona local](#)

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.

Configuração do cluster do Redis

ElastiCache para Redis, os clusters são compostos de 0 a 500 fragmentos (também chamados de grupos de nós). Os dados em um cluster Redis são particionados nos estilhaços no cluster. Seu aplicativo conecta-se a um cluster Redis usando um endereço de rede chamado de Endpoint. Os nós em um estilhaço do Redis cumprem uma das duas funções: uma primária de leitura/gravação e todos os outros nós secundários somente leitura (também chamados de réplicas de leitura). Além dos pontos de extremidade do nó, o cluster do Redis em si 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 ElastiCache para o Redis.



Para ter mais informações, consulte [Gerenciamento de clusters](#).

Requisitos de escalabilidade

Todos os clusters podem ser expandidos criando um novo cluster com o novo tipo de nó maior. Ao expandir um cluster do Redis, é possível propagá-lo de um backup e evitar que o novo cluster comece vazio.

Para obter mais informações, consulte [Dimensionamento ElastiCache para Redis](#) neste guia.

Requisitos de acesso

Por padrão, os ElastiCache clusters da Amazon são acessados a partir de instâncias do Amazon EC2. O acesso à rede a um ElastiCache cluster é limitado à conta que criou o cluster. Portanto, antes de poder acessar um cluster de uma instância do Amazon EC2, você deve autorizar a instância do Amazon EC2 a acessar o cluster. As etapas para essa autorização variam, dependendo de você ter executado na EC2-VPC ou no EC2-Classic.

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

Requisitos de região, zona de disponibilidade e zona local

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:

- [Escolher regiões e zonas de disponibilidade](#)
- [Uso de zonas locais com o ElastiCache](#)
- [Atenuar falhas](#)

Escolha do tamanho do nó

O tamanho do nó selecionado para o cluster afeta os custos, o desempenho e a tolerância a falhas.

Escolha do tamanho do nó

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

Responder às seguintes perguntas poderá ajudá-lo a determinar o tipo mínimo de nó de que você precisa para sua implementação do Redis.

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

Se for esse o caso e você estiver executando o Redis versão 5.0.6 ou superior, poderá obter melhor throughput e latência usando E/S aprimorada, em que as CPUs disponíveis são usadas para descarregar as conexões do cliente, em nome do mecanismo Redis. Se você estiver executando o Redis versão 7.0.4 ou superior, além de E/S aprimorada, você obterá aceleração adicional com multiplexação de E/S aprimorada, em que cada thread de E/S de rede dedicado canaliza comandos de vários clientes para o mecanismo Redis, aproveitando a capacidade da Redis de processar comandos em lotes com eficiência. No ElastiCache para Redis v7.1 e versões posteriores, estendemos a funcionalidade de threads de E/S melhorada para também lidar com a lógica da camada de apresentação. Por camada de apresentação, o que queremos dizer é que os threads de E/S aprimorados agora não estão apenas lendo a entrada do cliente, mas também analisando a entrada no formato de comando binário do Redis, que acaba sendo encaminhado para o thread principal para execução, proporcionando ganho de desempenho. Consulte a [publicação do blog](#) e a página de [versões compatíveis](#) para obter detalhes adicionais.

- Você tem workloads que acessam regularmente um pequeno percentual de seus dados?

Caso a resposta seja sim e seu sistema esteja funcionando no mecanismo Redis versão 6.2 ou posterior, você pode aproveitar a classificação de dados em níveis escolhendo o tipo de nó r6gd. Com a classificação de dados em níveis, os dados usados menos recentemente são armazenados em SSD. Quando eles são recuperados, há um pequeno custo de latência, que é equilibrado pela economia de custos. Para obter mais informações, consulte [Classificação de dados em níveis](#).

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 está executando?

Versões do Redis anteriores a 2.8.22 exigem que você reserve mais memória para failover, snapshot, sincronização e promoção de uma réplica para operações primárias. Esse requisito é importante porque você deve ter memória suficiente disponível para todas as gravações que ocorrem durante o processo.

O Redis versão 2.8.22 e posteriores usam um processo de salvamento sem bifurcação que requer menos memória disponível que o processo anterior.

Para ver mais informações, consulte:

- [Como a sincronização e o backup são implementados](#)
- [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#)
- Qual é a intensidade de gravação do seu aplicativo?

Gravar aplicativos pesados pode exigir significativamente mais memória disponível, memória não utilizada por dados, ao tirar snapshots ou fazer failover. Sempre que o processo BGSAVE for executado, você deve ter memória suficiente que não é usada pelos dados para acomodar todas as gravações que transpirem durante o processo BGSAVE. Exemplos são ao tirar um snapshot, ao sincronizar um cluster primário com uma réplica em um cluster e ao ativar o recurso de arquivo somente anexado (AOF). Outro exemplo acontece ao promover uma réplica para primária (se você tiver o Multi-AZ habilitado). O pior caso é quando todos os seus dados são reescritos durante o processo. Nesse caso, você precisa de um tamanho de instância de nó com o dobro da memória que é necessária para os dados isoladamente.

Para obter mais informações detalhadas, consulte [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#).

- Sua implementação será um cluster autônomo Redis (modo cluster desabilitado) ou um cluster Redis (modo cluster habilitado) com vários fragmentos?

Cluster do Redis (modo cluster desabilitado)

Se você estiver implementando um cluster Redis (modo cluster desabilitado), seu tipo de nó deve ser capaz de acomodar todos os seus dados mais a sobrecarga necessária, conforme descrito anteriormente.

Por exemplo, suponha que você estime que o tamanho total de todos os seus itens é de 12 GB. Nesse caso, você pode usar um nó cache .m3.xlarge com 13,3 GB de memória ou um nó cache .r3.large com 13,5 GB de memória. No entanto, talvez você precise de mais memória para operações BGSAVE. Se a sua aplicação for de escrita pesada, duplique os requisitos de memória para pelo menos 24 GB. Assim, use ou cache .m3.2xlarge com 27,9 GB de memória ou um cache .r3.xlarge com 30,5 GB de memória.

Redis (modo cluster habilitado) com vários fragmentos

Se você estiver implementando um cluster Redis (modo cluster habilitado) com vários fragmentos, o tipo de nó deve ser capaz de acomodar $\text{bytes-for-data-and-overhead} / \text{number-of-shards}$ bytes de dados.

Por exemplo, suponha que você estime que o tamanho total de todos os seus itens é de 12 GB e você tem dois fragmentos. Nesse caso, você pode usar um nó cache .m3.large com 6,05 GB de memória (12 GB/2). No entanto, talvez você precise de mais memória para operações BGSAVE. Se a sua aplicação for de escrita pesada, duplique os requisitos de memória para pelo menos 12 GB por fragmento. Assim, use ou cache .m3.xlarge com 13,3 GB de memória ou um cache .r3.large com 13,5 GB de memória.

- Você está usando Local Zones?

[Local Zones](#) (Zonas locais) permitem que você coloque recursos como um cluster do ElastiCache em vários locais próximos aos usuários. Mas, ao escolher o tamanho do nó, esteja ciente de que os tamanhos de nó disponíveis estão limitados ao seguinte no momento, independentemente dos requisitos de capacidade:

- Geração atual:

Tipos de nó 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 o uso da memória, a utilização do processador, o cache e as métricas de perdas no cache que são publicadas no CloudWatch. Você

pode notar que seu cluster não tem a taxa de acertos desejada ou que as chaves estão sendo removidas com muita frequência. Nesses casos, você pode escolher um tamanho de nó diferente com especificações de CPU e memória maiores.

Ao monitorar o uso da CPU, lembre-se que o Redis é de thread único. Assim, multiplique o uso da CPU relatado pelo número de núcleos da CPU para obter esse uso real. Por exemplo, uma CPU de quatro núcleos que informa uma taxa de uso de 20% é, na verdade, o único núcleo que o Redis está usando executando em 80%.

Criar um cluster

Os exemplos a seguir mostram como criar um cluster do Redis usando AWS Management Console a ElastiCache API AWS CLI e.

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

ElastiCache oferece suporte à replicação quando você usa o mecanismo Redis. Para monitorar a latência entre o momento em que os dados são gravados em um cluster primário de leitura/gravação do 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 em Tempo Universal Coordenado (UTC) atual. Como um cluster do Redis pode ser adicionado a um grupo de replicação mais tarde, essa chave está incluída em todos os clusters do 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 do Redis (modo de cluster desabilitado), siga as etapas em [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#).

Assim que o status do seu cluster for `available` (disponível), você poderá conceder acesso ao Amazon EC2 a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 4: conexão ao nó do cluster](#) e [Etapa 3: autorizar o acesso ao 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 [Excluir um cluster](#).

Criação de um cluster do Redis (modo cluster habilitado) (console)

Se estiver executando o Redis 3.2.4 ou posterior, poderá criar um cluster do Redis (modo cluster habilitado). Clusters Redis (modo cluster habilitado) oferecem suporte para o particionamento dos seus dados entre 1 e 500 fragmentos (API/CLI: grupos de nós), mas com algumas limitações. Para obter uma comparação entre Redis (modo cluster desabilitado) e Redis (modo cluster habilitado), consulte [Versões do ElastiCache para Redis com suporte](#).

Para criar um cluster Redis (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 Create VPC (Criar VPC) e siga as etapas em [Creating a Virtual Private Cloud \(VPC\)](#) (Criação de uma nuvem privada virtual (VPC)).
5. Na página do ElastiCache painel, escolha Criar cluster e, em seguida, escolha Criar cluster Redis.
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 Redis com ElastiCache 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 específicos do Redis](#) e [Criar um parameter group](#).

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 ter 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 ter mais informações, consulte [Classificação de dados em níveis](#).

- e. Em Number of shards (Número de fragmentos), escolha o número de fragmentos (partições/grupos de nós) desejado para este cluster do Redis (modo cluster habilitado).

Para algumas versões do Redis (modo cluster habilitado), é possível alterar dinamicamente o número de fragmentos no cluster:

- Redis 3.2.10 e posterior: se o cluster estiver executando o Redis 3.2.10 ou versões posteriores, será possível alterar dinamicamente o número de fragmentos no cluster. Para ter mais informações, consulte [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#).
- Outras versões do Redis: se o cluster estiver executando uma versão do Redis 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 ter mais informações, consulte [Restauração de um backup para um novo cache](#).

- f. Em Replicas per shard (Réplicas por fragmento), escolha o número de nós de réplica de leitura desejados em cada fragmento.

As restrições a seguir existem para o Redis (modo cluster habilitado).

- Se você tiver o Multi-AZ habilitado, verifique se tem pelo menos uma réplica por fragmento.
- O número de réplicas é o mesmo para cada fragmento ao criar o cluster usando o console.
- O número de réplicas de leitura por fragmento é fixo e não pode ser alterado. Se você achar que precisa de mais ou menos réplicas por fragmento (API/CLI: grupo de nós), deverá criar um novo cluster com o novo número de réplicas. Para ter mais informações, consulte [Propagação de um novo cluster autoprojeto 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 endereços IPv4 e IPv6 atribuídos a eles para operar no modo de pilha dupla e uma sub-rede somente IPv6 para operar somente como IPv6.

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

Selecione um Discovery IP type (Tipo de IP de descoberta). Somente os endereços IP do protocolo escolhido são retornados.

Para obter mais informações, consulte:

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

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

Para ter 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 ter mais informações, consulte [Escolher regiões e zonas de disponibilidade](#).

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

 Note

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

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

de usuários que possam acessar o cluster. Para ter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).

- Redis AUTH Default User (Usuário padrão Redis AUTH): um mecanismo de autenticação para o servidor Redis. Para obter mais informações, consulte [Redis AUTH](#).
- Redis AUTH (Redis AUTH: um mecanismo de autenticação para o servidor Redis. Para obter mais informações, consulte [Redis AUTH](#).

 Note

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

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

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

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

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

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

início e Duração (em horas) para sua janela de manutenção. Todos os horários são em UCT.

Para ter mais informações, consulte [Gerenciamento da manutenção](#).

9. (Opcional) Para Logs:

- Em Log format (Formato do log), escolha Text (Texto) ou JSON.
- Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
- Em Destino do registro, escolha Criar novo e insira o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose, ou escolha Selecionar existente e escolha o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose,

10. Para Tags, para ajudá-lo a gerenciar seus clusters e outros ElastiCache recursos, você pode atribuir seus próprios metadados a cada recurso na forma de tags. Para obter mais informações, consulte [Marcação de recursos do 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 Redis com ElastiCache Multi-AZ](#)
2. Siga as etapas em [Uso do Outposts](#).

Para criar o equivalente usando a ElastiCache API ou AWS CLI em vez do ElastiCache console, veja o seguinte:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Assim que o status do seu cluster for available, você poderá conceder acesso ao EC2 a ele, conectar-se a ele e começar a usá-lo. Para obter mais informações, consulte [Etapa 4: conexão ao nó do cluster](#) e [Etapa 3: autorizar o acesso ao 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 [Excluir um cluster](#).

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 [Excluir um cluster](#).

Criação de um cluster de Redis (modo de cluster desabilitado) (CLI)

Example Um cluster do Redis (modo cluster desabilitado) sem réplicas de leitura.

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

Note

Ao criar cluster usando um tipo de nó da família `r6gd`, é necessário repassar o parâmetro `data-tiering-enabled`.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--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
```

Criação de um cluster do Redis (modo cluster habilitado) (AWS CLI)

Clusters do Redis (modo cluster habilitado) (API/CLI: grupos de replicação) não podem ser criados usando a operação `create-cache-cluster`. Para criar um cluster do Redis (modo cluster habilitado) (API/CLI: grupo de replicação), consulte [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio \(AWS CLI\)](#).

Para obter mais informações, consulte o tópico AWS CLI de ElastiCache referência [create-replication-group](#).

Criação de um cluster (ElastiCache API)

Para criar um cluster usando a ElastiCache API, use a `CreateCacheCluster` ação.

Important

Assim que seu cluster se tornar disponível, você será cobrado por cada hora ou hora parcial em que ele estiver, mesmo que você não o esteja usando. Para interromper as cobranças aplicáveis para esse cluster, você deve excluí-lo. Consulte [Excluir um cluster](#).

Tópicos

- [Criação de um cluster de cache \(ElastiCache API\) Redis \(modo de cluster desativado\)](#)
- [Criação de um cluster de cache no Redis \(modo de cluster ativado\) \(ElastiCache API\)](#)

Criação de um cluster de cache (ElastiCache API) Redis (modo de cluster desativado)

O código a seguir cria um cluster de cache (ElastiCache API) do Redis (modo de cluster desativado).

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 Redis (modo de cluster ativado) (ElastiCache API)

Clusters do Redis (modo cluster habilitado) (API/CLI: grupos de replicação) não podem ser criados usando a operação `CreateCacheCluster`. Para criar um cluster do Redis (modo cluster habilitado) (API/CLI: grupo de replicação), consulte [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio \(API do ElastiCache\)](#).

Para obter mais informações, consulte o tópico de referência ElastiCache da API [CreateReplicationGroup](#).

Visualização dos detalhes de um cluster

Você pode visualizar informações detalhadas sobre um ou mais clusters usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Visualização de detalhes de um cluster do Redis (modo cluster desabilitado) (console)

Você pode visualizar os detalhes de um cluster do Redis (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

O procedimento a seguir detalha como visualizar os detalhes de um cluster do Redis (modo cluster desabilitado) usando o console do ElastiCache.

Para exibir os detalhes de um cluster do Redis (modo cluster desabilitado)

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel do console do ElastiCache, escolha Redis para exibir uma lista de todos os clusters que estão executando qualquer versão do Redis.
3. Para ver os detalhes de um cluster, marque a caixa de seleção à esquerda do nome do cluster. Certifique-se de selecionar um cluster que esteja executando o mecanismo Redis, não o Redis clusterizado. Ao fazer isso, os detalhes sobre o cluster serão exibidos, incluindo o endpoint principal do cluster.
4. Como visualizar informações do nó:
 - a. Escolha o nome do cluster.
 - b. Selecione a guia Shards and nodes (Fragmentos e nós) Ao fazer isso, os detalhes sobre cada nó serão exibidos, incluindo o endpoint do nó, que você precisa usar para ler do cluster.
5. Para visualizar as métricas, escolha a guia Metrics (Métricas), que exibe as métricas relevantes para todos os nós do cluster. Para obter mais informações, consulte [Monitoramento do uso com métricas do CloudWatch](#).
6. Para visualizar os logs, escolha a guia Logs, que indica se o cluster está usando logs lentos ou logs do mecanismo e fornece detalhes relevantes. Para obter mais informações, consulte [Entrega de logs](#).
7. Escolha a guia Network and security (Rede e segurança) para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-redes do cluster. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#).

8. Escolha a guia Maintenance (Manutenção) para ver detalhes sobre as configurações de manutenção do cluster. Para obter mais informações, consulte [Gerenciamento da manutenção](#).
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 [Marcação de recursos do ElastiCache](#).

Visualização de detalhes de um cluster do Redis (modo cluster habilitado) (console)

Você pode visualizar os detalhes de um cluster do Redis (modo cluster habilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

O procedimento a seguir detalha como visualizar os detalhes de um cluster do Redis (modo cluster habilitado) usando o console do ElastiCache.

Para exibir os detalhes de um cluster do Redis (modo cluster habilitado)

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS de interesse.
3. No painel do console do ElastiCache, escolha Redis para exibir uma lista de todos os clusters que estão executando qualquer versão do Redis.
4. Para ver os detalhes de um cluster do Redis (modo cluster habilitado), escolha a caixa de seleção à esquerda do nome do cluster. Certifique-se de escolher um cluster executando o mecanismo Clustered Redis, e não apenas o Redis.

A tela se expande abaixo do cluster e exibe detalhes sobre o cluster, incluindo o endpoint de configuração de cluster.

5. Para ver uma lista dos fragmentos do cluster e o número de nós em cada fragmento, escolha a guia Shards and nodes (Fragmentos e nós).
6. Como visualizar informações específicas sobre um nó:
 - Escolha o ID do fragmento.

Isso exibirá informações sobre cada nó, incluindo o endpoint de cada nó que é necessário usar para ler dados do cluster.

7. Para visualizar as métricas, escolha a guia Metrics (Métricas), que exibe as métricas relevantes para todos os nós do cluster. Para obter mais informações, consulte [Monitoramento do uso com métricas do CloudWatch](#).
8. Para visualizar os logs, escolha a guia Logs, que indica se o cluster está usando logs lentos ou logs do mecanismo e fornece detalhes relevantes. Para obter mais informações, consulte [Entrega de logs](#).
9. Escolha a guia Network and security (Rede e segurança) para ver detalhes sobre a conectividade de rede e a configuração do grupo de sub-rede do cluster, o grupo de segurança da VPC e qual método de criptografia, se houver, está habilitado no cluster. Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#) e [Segurança de dados no Amazon ElastiCache](#).
10. Escolha a guia Maintenance (Manutenção) para ver detalhes sobre as configurações de manutenção do cluster. Para obter mais informações, consulte [Gerenciamento da manutenção](#).
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 [Marcação de recursos do ElastiCache](#).

Visualização dos detalhes de um cluster (AWS CLI)

O seguinte código lista os detalhes de *my-cluster*:

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

Substitua *my-cluster* pelo nome de seu cluster em um caso em que o cluster seja criado com um nó de cache e nenhum fragmento usando o comando `create-cache-cluster`.

```
{
  "CacheClusters": [
    {
      "CacheClusterStatus": "available",
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ]
    }
  ]
}
```

```

    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "08:30-09:30",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false,
    "CacheClusterId": "my-cluster1",
    "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
    "PreferredAvailabilityZone": "us-west-2c",
    "AuthTokenEnabled": false,
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis3.2"
    },
    "SnapshotRetentionLimit": 0,
    "AutoMinorVersionUpgrade": true,
    "EngineVersion": "3.2.10",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
}

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
      "AuthTokenEnabled": false,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,

```

```

    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-002",

```

```

    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2"
    },
    "SnapshotRetentionLimit": 0,

```

```

    "EngineVersion": "3.2.10",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  }

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "AuthTokenEnabled": true,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
      "CacheClusterStatus": "available",
      "AtRestEncryptionEnabled": true,
      "PreferredAvailabilityZone": "us-west-2a",
      "TransitEncryptionEnabled": true,
      "ReplicationGroupId": "my-cluster3",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
      "CacheClusterId": "my-cluster3-0001-001",
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",
      "DataTiering": "disabled",
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
      },
      "SnapshotRetentionLimit": 0,
      "EngineVersion": "6.0",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1
    },
    {

```

```

    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": true,

```

```

    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",

```

```

    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",

```

```
    "CacheParameterGroupName": "default.redis3.2.cluster.on"
  },
  "SnapshotRetentionLimit": 0,
  "EngineVersion": "3.2.6",
  "CacheSecurityGroups": [],
  "NumCacheNodes": 1
},
{
  "SecurityGroups": [
    {
      "Status": "active",
      "SecurityGroupId": "sg-dbe93fa2"
    }
  ],
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
  "AuthTokenEnabled": true,
  "CacheSubnetGroupName": "default",
  "SnapshotWindow": "12:30-13:30",
  "AutoMinorVersionUpgrade": true,
  "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
  "CacheClusterStatus": "available",
  "AtRestEncryptionEnabled": true,
  "PreferredAvailabilityZone": "us-west-2a",
  "TransitEncryptionEnabled": true,
  "ReplicationGroupId": "my-cluster3",
  "Engine": "redis",
  "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
  "CacheClusterId": "my-cluster3-0002-003",
  "PendingModifiedValues": {},
  "CacheNodeType": "cache.r4.large",
  "DataTiering": "disabled",
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "ParameterApplyStatus": "in-sync",
    "CacheParameterGroupName": "default.redis6.x.cluster.on"
  },
  "SnapshotRetentionLimit": 0,
  "EngineVersion": "6.0",
  "CacheSecurityGroups": [],
  "NumCacheNodes": 1
}
]
```

```
}
```

Em um caso em que o cluster seja criado com o AWS Management Console (nó de cluster habilitado ou desabilitado com um ou mais fragmentos), use o comando a seguir para descrever os detalhes do cluster [substitua *my-cluster* pelo nome do grupo de replicação (nome de seu cluster)]:

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Para obter mais informações, consulte a AWS CLI para o tópico do ElastiCache [describe-cache-clusters](#).

Visualização dos detalhes de um cluster (API do ElastiCache)

Você pode visualizar os detalhes de um cluster usando a ação DescribeCacheClusters da API do ElastiCache. Se o parâmetro CacheClusterId estiver incluído, os detalhes do cluster especificado serão retornados. Se o parâmetro CacheClusterId for omitido, os detalhes para até MaxRecords (padrão 100) clusters serão retornados. O valor para MaxRecords não pode ser inferior a 20 ou superior a 100.

O código a seguir lista os detalhes para my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterId=my-cluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

O código a seguir lista os detalhes para até 25 clusters.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte o tópico de referência da API do ElastiCache [DescribeCacheClusters](#).

Modificando um cluster ElastiCache

Além de adicionar ou remover nós de um cluster, pode haver momentos em que você precisa fazer outras alterações em um cluster existente, como adicionar um grupo de segurança, alterar a janela de manutenção ou um grupo de parâmetros.

Recomendamos que você tenha sua janela de manutenção cair no momento da menor utilização. Assim, talvez seja necessário modificá-la de tempos em tempos.

Quando você altera parâmetros de um cluster, a alteração é aplicada no cluster imediatamente ou após ele ser reiniciado. Isso é verdadeiro se você alterar o próprio grupo de parâmetro do cluster ou um valor do parâmetro dentro do grupo do parâmetro do cluster. Para determinar quando uma mudança de parâmetro particular é aplicada, consulte a seção [As alterações terão efeito da coluna Detalhes nas tabelas de Parâmetros específicos do Redis](#).

Usando o 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 clientes Redis se conectem usando o modo de cluster ativado e o modo de cluster desativado. Depois de migrar todos os clientes Redis para usar o

modo de cluster ativado, você poderá 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 [Versões de mecanismos e atualização](#). No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

- Grupos de segurança da VPC
- Grupo do parâmetro
- Tipo de nó

Note

Se o cluster estiver usando um tipo de nó da família r6gd, você só poderá escolher um tamanho de nó diferente nessa família. Se você escolher um tipo de nó da família r6gd, a classificação de dados em níveis será ativada automaticamente. Para mais informações, consulte [Classificação de dados em níveis](#).

- Multi-AZ
- Failover automático (somente no modo cluster desabilitado)
- Habilitar backups automáticos
- ID do nó de backup
- Período de retenção de backup
- Janela de backup
- Tópico para notificação do SNS

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 no tipo de nó e na versão do mecanismo serão aplicadas durante a próxima janela de manutenção. Outras modificações, como a alteração da janela de manutenção, são aplicadas imediatamente.

7. Escolha Modificar.

Para habilitar/desabilitar a entrega de logs

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 Cluster details (Detalhes do cluster), escolha a guia Logs,
3. Para habilitar/desabilitar logs lentos, escolha Enable (Habilitar) ou Disable (Desabilitar).

Se você escolher habilitar:

- a. Em Log format (Formato do log), escolha JSON ou Text (Texto).
- b. Em Tipo de destino do registro, escolha CloudWatch Logs ou Kinesis Firehose.
- c. 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.
- d. Escolha Enable (Habilitar).

Para alterar sua configuração:

1. Selecione Modify (Modificar)
2. Em Log format (Formato do log), escolha JSON ou Text (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

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 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 [Versões de mecanismos e atualização](#). 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 a ElastiCache API

Você pode modificar um cluster existente usando a `ModifyCacheCluster` operação ElastiCache da API. Para modificar o valor de configuração de um cluster, especifique o ID do cluster, o parâmetro a ser alterado e o novo valor do parâmetro. O exemplo a seguir altera a janela de manutenção para um cluster chamado `my-cluster` e aplica a alteração imediatamente.

Important

Você pode atualizar para versões mais recentes do mecanismo. 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 [Versões de mecanismos e atualização](#). No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente e criando-o novamente.

As quebras de linha foram adicionadas para legibilidade.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&PreferredMaintenanceWindow=sun:23:00-mon:02:00  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150901T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150901T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

O parâmetro `ApplyImmediately` aplica-se apenas a modificações no tipo de nó, na versão do mecanismo e a alteração do número de nós em um cluster. Se quiser aplicar qualquer uma dessas alterações imediatamente, defina o parâmetro `ApplyImmediately` como `true`. Se preferir adiar essas alterações para sua próxima janela de manutenção, defina o parâmetro `ApplyImmediately` como `false`. Outras modificações, como a alteração da janela de manutenção, são aplicadas imediatamente.

Para obter mais informações, consulte o tópico de referência ElastiCache da API [ModifyCacheCluster](#).

Adição de nós a um cluster

Para reconfigurar o cluster do Redis (modo cluster habilitado), consulte [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#)

Você pode usar o ElastiCache Management Console, a AWS CLI ou a API do ElastiCache para adicionar nós ao seu cluster.

Usar a AWS Management Console

Se você quiser adicionar um nó a um cluster do Redis (modo cluster desabilitado) de nó único (sem replicação habilitada), o processo será de duas etapas: primeiro adicione a replicação e depois adicione um nó de réplica.

Tópicos

- [Para adicionar replicação a um cluster Redis sem estilhaços](#)
- [Para adicionar nós a um cluster \(console\)](#)

O procedimento a seguir adiciona replicação a um Redis de nó único que não possui replicação habilitada. Quando você adiciona replicação, o nó existente torna-se o nó primário no cluster habilitado para replicação. Depois que a replicação é adicionada, você pode adicionar até 5 nós de réplica ao cluster.

Para adicionar replicação a um cluster Redis sem estilhaços

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).

Uma lista de clusters que executam o mecanismo Redis é exibida.

3. Escolha o nome de um cluster (não a caixa à esquerda do nome do cluster) ao qual você quer adicionar nós.

O seguinte é verdade para um cluster Redis que não possui replicação habilitada:

- Ele está executando Redis, e não o Clustered Redis.
- Tem zero estilhaços.

Se o cluster tiver estilhaços, a replicação já estará ativada e você poderá continuar em [Para adicionar nós a um cluster \(console\)](#).

4. Escolha Add replication.
5. Em Adicionar replicação, insira uma descrição para esse cluster habilitado para replicação.
6. Escolha Add (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 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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha o mecanismo em execução no cluster ao qual você deseja adicionar nós.

É exibida uma lista de clusters que executam o mecanismo escolhido.

3. Na lista de clusters, no cluster ao qual você deseja adicionar um nó, escolha seu nome.

Se o cluster for um cluster do Redis (modo cluster habilitado), consulte [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#).

Se o cluster é um cluster do Redis (modo cluster habilitado) com zero fragmentos, primeiro conclua as etapas em [Para adicionar replicação a um cluster Redis sem estilhaços](#).

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	Excluir	Excluir	<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	Excluir	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	Excluir	<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; background-color: #fff9e6;"> <p>⚠ Important</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.

Usar a AWS CLI

Se quiser adicionar nós a um cluster do Redis (modo cluster habilitado) que não possua a replicação habilitada, você deverá primeiro criar o grupo de replicação especificando o cluster existente como o primário. Para obter mais informações, consulte [Criação de um grupo de replicação usando um cluster de cache 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 a AWS CLI, use a AWS CLI operação `increase-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: o ID do grupo de replicação ao qual você deseja adicionar nós.
- `--new-replica-count`: especifica o número de nós que você deseja neste grupo de replicação após a modificação ser aplicada. Para adicionar nós a esse cluster, `--new-replica-count` deve ser maior que o número atual de nós neste cluster.
- `--apply-immediately` ou `--no-apply-immediately`, que especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.

Para Linux, macOS ou Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 4 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 4 ^  
  --apply-immediately
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test",  
    "Status": "modifying",  
    "PendingModifiedValues": {},
```

```
"MemberClusters": [
  "node-test-001",
  "node-test-002",
  "node-test-003",
  "node-test-004",
  "node-test-005"
],
"NodeGroups": [
  {
    "NodeGroupId": "0001",
    "Status": "modifying",
    "PrimaryEndpoint": {
      "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "ReaderEndpoint": {
      "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "NodeGroupMembers": [
      {
        "CacheClusterId": "node-test-001",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "primary"
      },
      {
        "CacheClusterId": "node-test-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
      },
      {
        "CacheClusterId": "node-test-003",
```

```

        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    }
]
}
],
"SnapshottingClusterId": "node-test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
"DataTiering": "disabled",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
}
}

```

Para obter mais informações, consulte o tópico sobre a AWS CLI [increase-replica-count](#).

Uso da API do ElastiCache

Se quiser adicionar nós a um cluster do Redis (modo cluster desabilitado) que não possua a replicação habilitada, você deverá primeiro criar o grupo de replicação especificando o cluster existente como o primário. Para obter mais informações, consulte [Adição de réplicas a um cluster independente do Redis \(modo cluster desabilitado\) \(API do ElastiCache\)](#). Depois que o grupo de replicação estiver disponível, você poderá continuar com o processo a seguir.

Para adicionar nós a um cluster (API do ElastiCache)

- Chame a operação de API `IncreaseReplicaCount` com os seguintes parâmetros:
 - `ReplicationGroupId`: o ID do cluster ao qual você deseja adicionar os nós.

- **NewReplicaCount:** o parâmetro `NewReplicaCount` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada. Para adicionar nós a esse cluster, `NewReplicaCount` deve ser maior que o número atual de nós neste cluster. Se esse valor for inferior ao número atual de nós, use a API `DecreaseReplicaCount` com o número de nós a serem removidos do cluster.
- **ApplyImmediately** Especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.
- **Region** Especifica a região da AWS do cluster ao qual você deseja adicionar nós.

O exemplo a seguir mostra uma chamada para adicionar nós a um cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=true  
  &NumCacheNodes=4  
  &ReplicationGroupId=my-replication-group  
  &Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico da API do ElastiCache [IncreaseReplicaCount](#).

Remoção de nós de um cluster

Você pode excluir um nó de um cluster usando o AWS Management Console, a AWS CLI ou a API do ElastiCache.

Usar a AWS Management Console

Para remover nós de um cluster (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS do cluster da qual você deseja remover nós.
3. No painel de navegação, escolha o mecanismo em execução no cluster do qual você deseja remover um nó.

É exibida uma lista de clusters que executam o mecanismo escolhido.

4. Na lista de clusters, escolha o nome do cluster do qual você deseja remover um nó.

É exibida uma lista dos nós do cluster.

5. Escolha a caixa à esquerda do ID do nó para o nó que você deseja remover. Usando o console do ElastiCache, é possível excluir somente um nó de cada vez. Portanto, escolher vários nós significa que não é possível usar o botão Delete node (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

Se excluir o nó fizer com que o cluster não seja mais compatível com Multi-AZ, primeiro desmarque a caixa de seleção Multi-AZ e 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	Excluir	Excluir	<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	Excluir	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	Excluir	<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>
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>

Cenários	Operação pendente	Nova solicitação	Resultados
			<div style="border: 1px solid #f08080; padding: 10px; background-color: #fff9e6;"> <p>⚠ Important</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.

Usar a AWS CLI

1. Identifique os IDs dos nós que você deseja remover. Para obter mais informações, consulte [Visualização dos detalhes de um cluster](#).
2. Use a operação `decrease-replica-count` da CLI com uma lista dos nós a serem removidos, como no exemplo a seguir.

Para remover nós de um cluster usando a interface da linha de comando, use o comando `decrease-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: o ID do grupo de replicação do qual você deseja remover nós.
- `--new-replica-count`: o parâmetro `--new-replica-count` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada.
- `--replicas-to-remove`: uma lista de IDs de nós 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 região da AWS do cluster da qual você deseja remover nós.

Note

Você pode passar apenas um dos parâmetros `--replicas-to-remove` ou `--new-replica-count` ao chamar esta operação.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 2 \  
  --region us-east-2 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 3 ^  
  --region us-east-2 ^  
  --apply-immediately
```

Esta operação produz uma saída semelhante à seguinte (formato JSON):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test"  
  },  
  "Status": "modifying",  
  "PendingModifiedValues": {},  
  "MemberClusters": [  
    "node-test-001",  
    "node-test-002",  
    "node-test-003",  
    "node-test-004",  
    "node-test-005",  
    "node-test-006"  
  ],  
}
```

```
"NodeGroups": [  
  {  
    "NodeGroupId": "0001",  
    "Status": "modifying",  
    "PrimaryEndpoint": {  
      "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",  
      "Port": 6379  
    },  
    "ReaderEndpoint": {  
      "Address": "node-test-  
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",  
      "Port": 6379  
    },  
    "NodeGroupMembers": [  
      {  
        "CacheClusterId": "node-test-001",  
        "CacheNodeId": "0001",  
        "ReadEndpoint": {  
          "Address": "node-  
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "PreferredAvailabilityZone": "us-west-2a",  
        "CurrentRole": "primary"  
      },  
      {  
        "CacheClusterId": "node-test-002",  
        "CacheNodeId": "0001",  
        "ReadEndpoint": {  
          "Address": "node-  
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "PreferredAvailabilityZone": "us-west-2c",  
        "CurrentRole": "replica"  
      },  
      {  
        "CacheClusterId": "node-test-003",  
        "CacheNodeId": "0001",  
        "ReadEndpoint": {  
          "Address": "node-  
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        }  
      },  
    ]  
  },  
]
```

```
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-004",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-005",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "node-test-006",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    }
]
}
},
"SnapshottingClusterId": "node-test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
```

```
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
  "DataTiering": "disabled",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
  }
}
```

Alternativamente, você também poderia chamar o `decrease-replica-count` e, em vez de passar o parâmetro `--new-replica-count`, você pode passar o parâmetro `--replicas-to-remove`, como mostrado a seguir:

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id my-replication-group \
  --replicas-to-remove node-test-003 \
  --region us-east-2 \
  --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id my-replication-group ^
  --replicas-to-remove node-test-003 ^
  --region us-east-2 ^
  --apply-immediately
```

Para obter mais informações, consulte os tópicos da AWS CLI [decrease-replica-count](#).

Uso da API do ElastiCache

Para remover nós usando a API do ElastiCache, chame a operação da API

`DecreaseReplicaCount` com o ID do grupo de replicação e uma lista de nós para remoção, 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 região da AWS do cluster da qual você deseja remover um nó.

O exemplo a seguir remove imediatamente os nós 0004 e 0005 do cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DecreaseReplicaCount  
  &ReplicationGroupId=my-replication-group  
  &ApplyImmediately=true  
  &ReplicasToRemove=node-test-003  
  &Region us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico da API do ElastiCache [DecreaseReplicaCount](#).

Cancelamento de operações de adição ou exclusão de nós pendentes

Se você tiver optado por não aplicar uma alteração imediatamente, a operação terá o status pending até que seja realizada na sua 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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a região da AWS na qual 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.

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

Ao excluir um cluster do ElastiCache for Redis, seus snapshots manuais são retidos. Também é possível criar um snapshot final antes que o cluster seja excluído. Os snapshots de cache automáticos não são retidos.

Usar a 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 console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de console do ElastiCache, escolha o mecanismo que o cluster que você deseja excluir está executando.

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 de cada vez no console do ElastiCache. Escolher vários clusters desabilita a operação de exclusão.

4. Em Actions, escolha Delete.
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.

Usar a AWS CLI

O código a seguir exclui o cluster de cache `my-cluster`.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

A ação `delete-cache-cluster` da CLI apenas exclui 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 a AWS CLI para o tópico do ElastiCache [delete-cache-cluster](#).

Uso da API do ElastiCache

O código a seguir exclui o cluster `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteCacheCluster  
&CacheClusterId=my-cluster  
&Region us-east-2  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
```

```
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150202T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

A operação `DeleteCacheCluster` da API apenas exclui 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 de referência da API do ElastiCache [DeleteCacheCluster](#).

Acesso ao seu cluster ou grupo de replicação

Suas instâncias do Amazon ElastiCache são projetadas para acesso por meio de uma instância do Amazon EC2.

Se você iniciou sua instância do ElastiCache em uma Amazon Virtual Private Cloud (Amazon VPC), poderá acessar sua instância do ElastiCache a partir de uma instância do Amazon EC2 na mesma Amazon VPC. Ou, usando o emparelhamento da VPC, você pode acessar sua instância do ElastiCache de um Amazon EC2 em uma Amazon VPC diferente.

Se você iniciou sua instância do ElastiCache no EC2-Classic, você permitirá que a instância do EC2 acesse seu cluster, concedendo ao grupo de segurança do Amazon EC2 associado acesso de instância ao seu grupo de segurança do 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 iniciou seu cluster em EC2-VPC

Se você iniciou seu cluster em uma Amazon Virtual Private Cloud (Amazon VPC), poderá se conectar ao seu cluster do ElastiCache somente a partir de uma instância do Amazon EC2 que esteja sendo executada na mesma Amazon VPC. Nesse caso, você precisará conceder entrada de rede ao cluster.

Note

Se você estiver usando Local Zones (Zonas locais), verifique se você as habilitou. Para obter mais informações, consulte [Habilitar zonas locais](#). Ao fazer isso, sua VPC é estendida para essa zona local e sua VPC tratará a sub-rede como qualquer sub-rede em qualquer outra zona de disponibilidade, e gateways relevantes, tabelas de rota e outras considerações de grupos de segurança serão ajustados automaticamente.

Para conceder entrada na rede de um grupo de segurança da Amazon VPC para um cluster

1. Faça login no AWS Management Console e abra o console Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação, em Network & Security, escolha Security Groups.
3. Na lista de grupos de segurança, escolha o de segurança para a sua Amazon VPC. A menos que você tenha criado um grupo de segurança para uso com o ElastiCache, esse grupo de segurança será chamado de default.
4. Escolha a guia Inbound e faça o seguinte:
 - a. Selecione a opção Editar.
 - b. Escolha Adicionar regra.
 - c. Na coluna Tipo, escolha Regra TCP personalizada.
 - d. Na caixa Port range, digite o número da porta para o nó do cluster. Esse número deve ser o mesmo que você especificou quando você executou o cluster. A porta padrão do Redis é **6379**.
 - e. Na caixa Source (Fonte), escolha Anywhere (Qualquer lugar), que tem o intervalo de porta (0.0.0.0/0) para que qualquer instância do Amazon EC2 que você inicie na sua Amazon VPC possa se conectar aos seus nós do ElastiCache.

 Important

Abrir o cluster do ElastiCache para 0.0.0.0/0 não expõe o cluster à Internet, pois ele não possui um endereço IP público e, portanto, não pode ser acessado de fora da VPC. No entanto, o grupo de segurança padrão pode ser aplicado a outras instâncias do Amazon EC2 na conta do cliente, e essas instâncias podem ter um endereço IP público. Se eles estiverem executando algo na porta padrão, esse serviço poderá ser exposto involuntariamente. Portanto, recomendamos criar um grupo de segurança de VPC que será usado exclusivamente pelo ElastiCache. Para obter mais informações, consulte [Grupos de segurança personalizados](#).

- f. Escolha Save (Salvar).

Quando você ativa uma instância do Amazon EC2 na sua Amazon VPC, essa instância poderá se conectar ao seu cluster do ElastiCache.

Acesso a recursos do ElastiCache de fora da AWS

O Amazon ElastiCache é um serviço da AWS que oferece armazenamento de chave-valor na memória com base na nuvem. O serviço foi projetado para ser acessado exclusivamente pela AWS. No entanto, se o cluster do ElastiCache estiver hospedado dentro de uma VPC, será possível usar uma instância de conversão de endereços de rede (NAT) para oferecer acesso externo.

Requisitos

Os seguintes requisitos devem ser atendidos para você acessar os recursos do ElastiCache fora da AWS:

- O cluster deve residir em uma VPC e ser acessado por meio de uma instância NAT (Conversão de endereços de rede). Não existem exceções para esse requerimento.
- A instância NAT deve ser executada na mesma VPC que o cluster.
- A instância NAT deve ser iniciada em uma sub-rede pública separada do cluster.
- Um endereço IP elástico (EIP) deve ser associado à instância NAT. O recurso de encaminhamento de porta de iptables é usado para encaminhar uma porta na instância NAT à porta do nó de cache na VPC.

Considerações

As considerações a seguir devem ser mantidas em mente durante o acesso aos recursos do ElastiCache por fora do ElastiCache.

- Os clientes se conectam ao EIP e à porta de cache da instância NAT. O encaminhamento de porta na instância NAT direciona o tráfego ao nó do cluster 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 instância NAT 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 instância NAT.

- O tráfego dos clientes para a instância NAT não é criptografado. Portanto, você deve evitar o envio de dados confidenciais através da instância NAT.
- A instância NAT acrescenta a sobrecarga de manter outra instância.
- A instância NAT serve como um único ponto de falha. Para obter informações sobre como configurar o NAT de alta disponibilidade na VPC, consulte [Alta disponibilidade para instâncias NAT da Amazon VPC: um exemplo](#).

Como acessar recursos do ElastiCache por fora da AWS

O procedimento a seguir demonstra como se conectar aos recursos do ElastiCache usando uma instância NAT.

Estas etapas assumem o seguinte:

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

Em seguida, você precisa da NAT na direção oposta:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

Você também precisa habilitar o encaminhamento de IP, que permanece desabilitado por padrão:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf  
sudo sysctl --system
```

- Você está acessando um cluster Redis com:
 - Endereço IP: 10.0.1.230
 - Porta Redis padrão: 6379
 - Grupo de segurança: sg-bd56b7da
 - Endereço IP da instância da AWS: sg-bd56b7da
- Seu cliente confiável possui o endereço IP 198.51.100.27.
- Sua instância NAT possui o endereço IP elástico 203.0.113.73.
- Sua instância NAT tem o security group sg-ce56b7a9.

Para se conectar aos recursos do ElastiCache usando uma instância NAT

1. Crie uma instância NAT na mesma VPC que o seu cluster de cache, mas em uma sub-rede pública.

Por padrão, o assistente da VPC iniciará um tipo de nó cache.m1.small. Você deve selecionar um tamanho de nó conforme as suas necessidades. Você deve usar a AMI NAT EC2 para conseguir acessar o ElastiCache por fora da AWS.

Para obter informações sobre como criar uma instância NAT, consulte [NAT Instances](#) no Guia do usuário da AWS VPC.

2. Crie regras de security group para o cluster de cache e a instância NAT.

O grupo de segurança da instância NAT e a instância do cluster devem ter as seguintes regras:

- Duas regras de entrada
 - Uma para permitir conexões TCP de clientes confiáveis com cada porta de cache reencaminhada da instância NAT (6379 - 6381).
 - Uma segunda para permitir acesso SSH a clientes confiáveis.

Grupo de segurança da instância NAT: regras de entrada

Tipo	Protocolo	Intervalo de portas	Origem
Regra personalizada de TCP	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Uma regra de saída para permitir conexões TCP para a porta de cache (6379).

Grupo de segurança da instância NAT: regra de saída

Tipo	Protocolo	Intervalo de portas	Destino
Regra personalizada de TCP	TCP	6379	sg-ce56b7a9 (grupo de segurança da instância do cluster)

- Uma regra de entrada para o security group do cluster que permite conexões TCP da instância NAT com a porta de cache (6379).

Grupo de segurança da instância do cluster: regra de entrada

Tipo	Protocolo	Intervalo de portas	Origem
Regra personalizada de TCP	TCP	6379	sg-bd56b7da (grupo de segurança do cluster)

3. Valide as regras.

- Confirme se o cliente confiável é capaz de fazer SSH para a instância NAT.
- Confirme se o cliente confiável pode se conectar ao cluster a partir da instância NAT.

4. Adicione uma regra iptables à instância NAT.

Uma regra iptables deve ser adicionada à tabela NAT para cada nó no cluster para encaminhar a porta de cache da instância NAT ao nó do cluster. Um exemplo pode ser o seguinte:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to 10.0.1.230:6379
```

O número da porta deve ser exclusivo para cada nó no cluster. Por exemplo, se estiver trabalhando com um cluster Redis de três nós usando as portas 6379 - 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
```

5. Confirme se o cliente confiável pode se conectar ao cluster.

O cliente confiável deve se conectar ao EIP associado à instância NAT e à porta de cluster correspondente ao nó de cluster apropriado. Por exemplo, a string de conexão para o PHP pode ser o seguinte:

```
redis->connect( '203.0.113.73', 6379 );  
redis->connect( '203.0.113.73', 6380 );  
redis->connect( '203.0.113.73', 6381 );
```

Um cliente telnet também pode ser usado para verificar a conexão. Por exemplo:

```
telnet 203.0.113.73 6379  
telnet 203.0.113.73 6380  
telnet 203.0.113.73 6381
```

6. 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 cache do ElastiCache em um Amazon VPC](#)
- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente](#)
- [Instâncias NAT](#)
- [Configuração de clientes do ElastiCache](#)
- [Alta disponibilidade para instâncias NAT da Amazon VPC: um exemplo](#)

Localização de endpoints de conexão

Seu aplicativo conecta-se ao seu cluster usando endpoints. Um endpoint é o endereço exclusivo de um nó ou cluster.

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.

Quais endpoints usar

- Nó autônomo Redis, use o endpoint do nó para operações de leitura e gravação.
- Clusters do Redis (modo cluster desabilitado), use o Endpoint primário para todas as operações de gravação. Use o Endpoint de leitor para dividir uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura. Use os Endpoints de nó individuais para operações de leitura (na API/CLI, eles são chamados de Endpoints de leitura).
- Clusters do Redis (modo cluster habilitado), use o Endpoint de configuração do cluster para todas as operações com suporte a comandos do modo cluster habilitado. Você deve usar um cliente que ofereça suporte ao cluster do Redis (Redis 3.2). Você ainda pode ler de endpoints de nó individuais (na API/CLI, eles se chamam Endpoints de leitura).

As seções a seguir orientam pelo processo de descoberta dos endpoints necessários para o mecanismo que você está executando.

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

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

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

Um endpoint de leitor divide uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura em um cluster do ElastiCache para Redis. Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar várias réplicas de leitura do seu cluster do ElastiCache para Redis em diferentes zonas de disponibilidade da AWS (AZ) para garantir a alta disponibilidade dos endpoints leitores.

Note

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

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

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

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).

A tela de clusters aparecerá com uma lista de clusters do Redis (modo cluster desabilitado) e do Redis (modo cluster habilitado).

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

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

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

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

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

<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

Endpoint de nó para um cluster do Redis (modo cluster desabilitado)

- Como copiar um endpoint para a área de transferência:
 - Um endpoint por vez, localize o endpoint que você deseja copiar.
 - Escolha o ícone de cópia diretamente na frente do endpoint.

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

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

Criptografia em trânsito não habilitada

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

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

Criptografia em trânsito habilitada

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

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

Localização de endpoints para um cluster do Redis (modo cluster habilitado) (console)

O cluster do Redis (modo cluster habilitado) possui um único endpoint de configuração. Ao se conectar ao endpoint de configuração, a aplicação pode descobrir os endpoints primários e os endpoints de leitura para cada fragmento no cluster.

Para encontrar um endpoint de cluster do Redis (modo cluster habilitado)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Redis).

A tela de clusters aparecerá com uma lista de clusters do Redis (modo cluster desabilitado) e do Redis (modo cluster habilitado). Escolha o cluster do Redis (modo cluster habilitado) 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.

Localizar endpoints (AWS CLI)

Você pode usar a AWS CLI do Amazon ElastiCache para descobrir os endpoints para nós, clusters e grupos de replicação.

Tópicos

- [Encontrar endpoints para nós e clusters \(AWS CLI\)](#)
- [Encontrar os endpoints para grupos de replicação \(AWS CLI\)](#)

Encontrar endpoints para nós e clusters (AWS CLI)

Você pode usar a AWS CLI para descobrir os endpoints para um cluster e seus nós com o comando `describe-cache-clusters`. Para clusters do Redis, o comando retorna o endpoint do cluster. 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 as informações de cluster para o cluster do Redis (modo cluster desabilitado) de nó único mycluster.

Important

O parâmetro `--cache-cluster-id` pode ser usado com um ID de cluster do Redis (modo cluster desabilitado) de nó único ou IDs de nó específicos em grupos de replicação do Redis. O `--cache-cluster-id` de um grupo de replicação do Redis é um valor de 4 dígitos, como `0001`. Se `--cache-cluster-id` for o ID de um cluster (nó) em um grupo de replicação do Redis, o `replication-group-id` será incluído na saída.

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws elasticache describe-cache-clusters ^
```

```
--cache-cluster-id redis-cluster ^  
--show-cache-node-info
```

O resultado da operação acima deve ser semelhante a este (formato JSON).

```
{  
  "CacheClusters": [  
    {  
      "CacheClusterStatus": "available",  
      "SecurityGroups": [  
        {  
          "SecurityGroupId": "sg-77186e0d",  
          "Status": "active"  
        }  
      ],  
      "CacheNodes": [  
        {  
          "CustomerAvailabilityZone": "us-east-1b",  
          "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",  
          "CacheNodeStatus": "available",  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Address": "redis-cluster.amazonaws.com",  
            "Port": 6379  
          },  
          "ParameterGroupStatus": "in-sync"  
        }  
      ],  
      "AtRestEncryptionEnabled": false,  
      "CacheClusterId": "redis-cluster",  
      "TransitEncryptionEnabled": false,  
      "CacheParameterGroup": {  
        "ParameterApplyStatus": "in-sync",  
        "CacheNodeIdsToReboot": [],  
        "CacheParameterGroupName": "default.redis3.2"  
      },  
      "NumCacheNodes": 1,  
      "PreferredAvailabilityZone": "us-east-1b",  
      "AutoMinorVersionUpgrade": true,  
      "Engine": "redis",  
      "AuthTokenEnabled": false,  
      "PendingModifiedValues": {},  
      "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",  
    }  
  ]  
}
```

```
    "CacheSecurityGroups": [],
    "CacheSubnetGroupName": "default",
    "CacheNodeType": "cache.t2.small",
    "DataTiering": "disabled"
    "EngineVersion": "3.2.10",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
  }
]
}
```

Para obter mais informações, consulte o tópico [describe-cache-clusters](#).

Encontrar os endpoints para grupos de replicação (AWS CLI)

Você pode usar a AWS CLI para descobrir os endpoints para um grupo de replicação e seus clusters com o comando `describe-replication-groups`. O comando retorna o endpoint primário do grupo de replicação e uma lista de todos os clusters (nós) no grupo de replicação com seus endpoints, junto com o endpoint de leitor.

A seguinte operação recupera o endpoint primário e o endpoint de leitor para o grupo de replicação `myreplgroup`. Use o endpoint primário para todas as operações de gravação.

```
aws elasticache describe-replication-groups \
  --replication-group-id myreplgroup
```

Para Windows:

```
aws elasticache describe-replication-groups ^
  --replication-group-id myreplgroup
```

O resultado dessa operação deve ser semelhante a este (formato JSON).

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "test",
      "NodeGroups": [
        {
          "Status": "available",
```

```
    "NodeGroupMembers": [
      {
        "CurrentRole": "primary",
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "myreplgroup-001.amazonaws.com"
        },
        "CacheClusterId": "myreplgroup-001"
      },
      {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "myreplgroup-002.amazonaws.com"
        },
        "CacheClusterId": "myreplgroup-002"
      },
      {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "myreplgroup-003.amazonaws.com"
        },
        "CacheClusterId": "myreplgroup-003"
      }
    ],
    "NodeGroupId": "0001",
    "PrimaryEndpoint": {
      "Port": 6379,
      "Address": "myreplgroup.amazonaws.com"
    },
    "ReaderEndpoint": {
      "Port": 6379,
      "Address": "myreplgroup-ro.amazonaws.com"
    }
  },
  "ReplicationGroupId": "myreplgroup",
```

```
    "AutomaticFailover": "enabled",
    "SnapshottingClusterId": "myreplgroup-002",
    "MemberClusters": [
      "myreplgroup-001",
      "myreplgroup-002",
      "myreplgroup-003"
    ],
    "PendingModifiedValues": {}
  }
]
```

Para obter mais informações, consulte [describe-replication-groups](#) na Referência de comandos da AWS CLI.

Localização de endpoints (API do ElastiCache)

Você pode usar a API do Amazon ElastiCache para descobrir os endpoints para nós, clusters e grupos de replicação.

Tópicos

- [Localização de endpoints para nós e clusters \(API do ElastiCache\)](#)
- [Localização de endpoints para grupos de replicação \(API do ElastiCache\)](#)

Localização de endpoints para nós e clusters (API do ElastiCache)

Você pode usar a API do ElastiCache para descobrir os endpoints para um cluster e seus nós com a ação `DescribeCacheClusters`. Para clusters do Redis, o comando retorna o endpoint do cluster. Se você incluir o parâmetro opcional `ShowCacheNodeInfo`, a ação também retornará os endpoints dos nós individuais no cluster.

Example

```
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>
```

Localização de endpoints para grupos de replicação (API do ElastiCache)

Você pode usar a API do ElastiCache para descobrir os endpoints para um grupo de replicação e seus cluster com a ação `DescribeReplicationGroups`. A ação retorna o endpoint primário do grupo de replicação e uma lista de todos os clusters no grupo de replicação com seus endpoints, junto com o endpoint de leitor.

A operação a seguir recupera o endpoint primário (`PrimaryEndpoint`), o endpoints de leitor (`ReaderEndpoint`) e os endpoints de nós individuais (`ReadEndpoint`) para o grupo de replicação `myreplgroup`. Use o endpoint primário para todas as operações de gravação.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DescribeReplicationGroups
&ReplicationGroupId=myreplgroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

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

Operação com fragmentos

Um fragmento (API/CLI: grupo de nós) é uma coleção de um a seis nós do Redis. Um cluster do Redis (modo cluster desabilitado) 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 nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do 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 as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para ter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

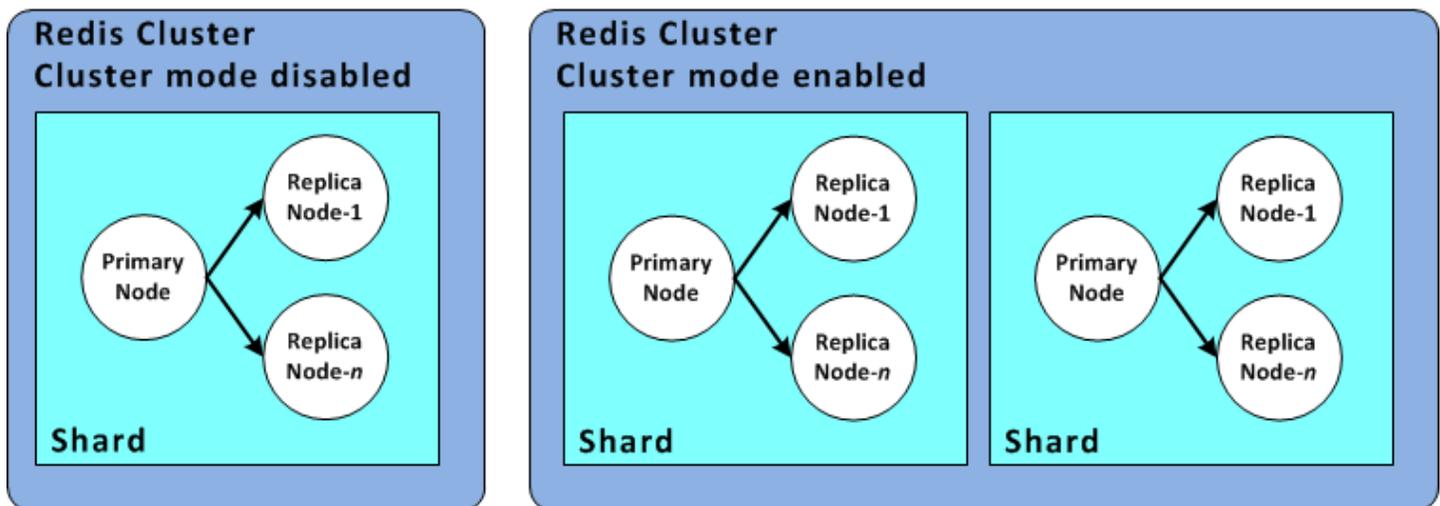
Para solicitar um aumento de limite, consulte [Limites do serviço da AWS](#) e escolha o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

Ao criar um cluster Redis (modo de cluster ativado) usando o ElastiCache console, você especifica o número de fragmentos no cluster e o número de nós nos fragmentos. Para ter mais informações,

consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#). Se você usar a ElastiCache API ou AWS CLI criar um cluster (chamado de grupo de replicação na API/CLI), poderá configurar o número de nós em um fragmento (API/CLI: grupo de nós) de forma independente. Para obter mais informações, consulte:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Cada nó em um fragmento tem as mesmas especificações de computação, armazenamento e memória. A ElastiCache API permite controlar atributos abrangentes, como o número de nós, as configurações de segurança e as janelas de manutenção do sistema.



Configurações de estilhaços do Redis

Para obter mais informações, consulte [Refragmentação e rebalanceamento de fragmentos online para o Redis \(modo cluster habilitado\)](#) e [Refragmentação e rebalanceamento de fragmentos offline para o Redis \(modo cluster habilitado\)](#).

Localização de um ID de fragmento

Você pode encontrar o ID de um fragmento usando o AWS Management Console, o AWS CLI ou a ElastiCache API.

Usando o AWS Management Console

Tópicos

- [Para Redis \(modo cluster desabilitado\)](#)

- [Para o Redis \(Modo cluster habilitado\)](#)

Para Redis (modo cluster desabilitado)

Os IDs de fragmento do grupo de replicação do Redis (modo cluster desabilitado) são sempre 0001.

Para o Redis (Modo cluster habilitado)

O procedimento a seguir usa o AWS Management Console para encontrar o ID do fragmento do grupo de replicação do Redis (modo de cluster ativado).

Para localizar o ID do fragmento em um grupo de replicação do Redis (modo cluster habilitado)

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 Redis e o nome do grupo de replicação do Redis (modo cluster habilitado) para o qual deseja encontrar os IDs do fragmento.
3. Na coluna Shard Name (Nome do fragmento), o ID do fragmento tem os últimos quatro dígitos do nome do fragmento.

Usando o AWS CLI

Para encontrar IDs de fragmento (grupo de nós) para grupos de replicação Redis (modo de cluster desativado) ou Redis (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 estilhaço (grupo de nós) estão *realçados* aqui para facilitar a localização deles.

```
{  
  "ReplicationGroups": [  
    {  
      "Status": "available",  
      "Description": "2 shards, 2 nodes (1 + 1 replica)",  
      "NodeGroups": [  
        {  
          "Status": "available",  
          "Slots": "0-8191",  
          "NodeGroupId": "0001",  
          "NodeGroupMembers": [  
            {  
              "PreferredAvailabilityZone": "us-west-2c",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0001-001"  
            },  
            {  
              "PreferredAvailabilityZone": "us-west-2a",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0001-002"  
            }  
          ]  
        },  
        {  
          "Status": "available",  
          "Slots": "8192-16383",  
          "NodeGroupId": "0002",  
          "NodeGroupMembers": [  
            {  
              "PreferredAvailabilityZone": "us-west-2b",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0002-001"  
            },  
            {  
              "PreferredAvailabilityZone": "us-west-2a",
```

```

        "CacheNodeId": "0001",
        "CacheClusterId": "sample-repl-group-0002-002"
    }
  ]
},
"ConfigurationEndpoint": {
  "Port": 6379,
  "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
},
"ClusterEnabled": true,
"ReplicationGroupId": "sample-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "13:00-14:00",
"MemberClusters": [
  "sample-repl-group-0001-001",
  "sample-repl-group-0001-002",
  "sample-repl-group-0002-001",
  "sample-repl-group-0002-002"
],
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled",
"PendingModifiedValues": {}
}
]
}

```

Usando a ElastiCache API

Para encontrar IDs de fragmento (grupo de nós) para grupos de replicação Redis (modo de cluster desativado) ou Redis (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. Caso esse parâmetro seja omitido, os detalhes de até **xxx** grupos de replicação são retornados.

Example

Esse comando retorna os detalhes do `sample-repl-group`.

Para Linux, macOS ou Unix:

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroup  
&ReplicationGroupId=sample-repl-group  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Comparação dos caches autoprotetados do Memcached e do Redis

O Amazon ElastiCache oferece suporte aos mecanismos de cache Memcached e Redis. Cada mecanismo oferece algumas vantagens. Use as informações contidas neste tópico para ajudá-lo a escolher o mecanismo e a versão que melhor atendam aos seus requisitos.

Important

Depois de criar um cluster de cache ou grupo de replicação, você pode atualizar para uma versão de mecanismo mais recente, mas não pode fazer downgrade para uma versão de mecanismo mais antiga. Se quiser usar uma versão de mecanismo mais antiga, deverá excluir o cluster de cache ou grupo de replicação existente e criá-lo novamente com a versão anterior do mecanismo.

Na superfície, os mecanismos parecem semelhantes. Cada um deles é um armazenamento de chaves/valores na memória. No entanto, na prática, existem diferenças significativas.

Escolha o Memcached se o seguinte se aplicar a você:

- Você precisa do modelo mais simples possível.
- Você precisa executar nós grandes com vários cores ou threads.
- Você precisa da capacidade de expandir e reduzir, adicionar e remover nós à medida que a demanda no seu sistema aumenta e diminui.
- Você precisa armazenar objetos em cache.

Escolha o Redis com uma versão do ElastiCache for Redis se o seguinte se aplicar a você:

- ElastiCache para Redis versão 7.0 (aprimorado)

Use [Funções do Redis](#), [Pub/Sub fragmentado](#) ou [Melhorias de ACL para Redis](#). Para obter mais informações, consulte [Redis versão 7.0 \(aprimorada\)](#).

- ElastiCache for Redis versão 6.2 (aprimorado)

Você deseja ter a capacidade de classificar dados em níveis entre memória e SSD usando o tipo de nó r6gd. Para mais informações, consulte [Classificação de dados em níveis](#).

- ElastiCache for Redis versão 6.0 (aprimorado)

Você deseja autenticar usuários com o controle de acesso baseado em função.

Para mais informações, consulte [Redis versão 6.0 \(aprimorada\)](#).

- ElastiCache for Redis versão 5.0.0 (aprimorado)

Você quer usar [Fluxos Redis](#), uma estrutura de dados de log que permite aos produtores anexarem novos itens em tempo real e também permite que os consumidores consumam mensagens de forma bloqueante ou não bloqueante.

Para obter mais informações, consulte [Redis versão 5.0.0 \(aprimorada\)](#).

- ElastiCache for Redis versão 4.0.10 (aprimorado)

Oferece suporte à criptografia e à adição ou remoção de fragmentos do cluster do Redis (modo cluster habilitado) de modo dinâmico.

Para obter mais informações, consulte [Redis versão 4.0.10 \(aprimorada\)](#).

As versões a seguir estão obsoletas ou chegaram ou estão prestes a chegar ao fim da vida útil.

- ElastiCache for Redis versão 3.2.10 (aprimorado)

Oferece suporte à adição ou remoção de fragmentos do cluster do Redis (modo cluster habilitado) de modo dinâmico.

 Important

Atualmente, o ElastiCache for Redis 3.2.10 não oferece suporte à criptografia.

Para ver mais informações, consulte:

- [Redis versão 3.2.10 \(aprimorada\)](#)
- Reestilhecimento online de melhores práticas para o Redis. Para obter mais informações, consulte o seguinte:
 - [Práticas recomendadas: reestilhecimento online](#)
 - [Refragmentação e rebalanceamento de fragmentos online para o Redis \(modo cluster habilitado\)](#)
- Para obter mais informações sobre como dimensionar clusters Redis, consulte [Escalabilidade](#).
- ElastiCache for Redis versão 3.2.6 (aprimorado)

Se você precisar da funcionalidade de versões anteriores do Redis, bem como dos recursos a seguir, escolha o ElastiCache for Redis 3.2.6:

- Criptografia em trânsito. Para obter mais informações, consulte [Criptografia em trânsito do Amazon ElastiCache for Redis](#).
- Criptografia em repouso. Para obter mais informações, consulte [Criptografia em repouso do Amazon ElastiCache for Redis](#).
- ElastiCache for Redis (modo cluster habilitado) versão 3.2.4

Se precisar da funcionalidade do Redis 2.8.x, bem como dos recursos a seguir, escolha o Redis 3.2.4 (modo de cluster):

- Você precisa dividir seus dados entre dois a 500 grupos de nós (somente no modo de cluster).
- Você precisa de indexação geoespacial (modo de cluster ou modo sem cluster).
- Você não precisa comportar vários bancos de dados.
- ElastiCache for Redis (modo sem cluster) 2.8.x e 3.2.4 (aprimorado)

Se o seguinte se aplicar a você, escolha o Redis 2.8.x ou o Redis 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.
- Você precisa oferecer suporte para vários bancos de dados.

Resumo da comparação do Memcached, Redis (modo cluster desabilitado) e Redis (modo cluster habilitado)

	Memcached	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
Engine versions+	1.4.5 and later	4.0.10 and later	4.0.10 and later
Data types	Simple	2.8.x - Complex * Complex	3.2.x and later - Complex
Data partitioning	Yes	No	Yes
Cluster is modifiable	Yes	Yes	3.2.10 and later - Limited
Online resharding	No	No	3.2.10 and later
Encryption	in-transit 1.6.12 and later	4.0.10 and later	4.0.10 and later
Data tiering	No	6.2 and later	6.2 and later
Certificações de conformidade			
Compliance Certifica tion			
FedRAMP	Sim, 1.6.12 e posterior	4.0.10 e posterior 4.0.10 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		
Multi-threaded	Yes	No	No
Node type upgrade	No	Yes	Yes
Engine upgrading	Yes	Yes	Yes

	Memcached	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
High availability (replication)	No	Yes	Yes
Automatic failover	No	Optional	Required
Pub/Sub capabilities	No	Yes	Yes
Sorted sets	No	Yes	Yes
Backup and restore	No	Yes	Yes
Geospatial indexing	No	4.0.10 and later	Yes

Observações:

string, objects (like databases)

* string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog

string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes

+ Excludes versions which are deprecated, have reached or soon to reach end of life.

Depois de escolher o mecanismo do seu cluster, recomendamos que você use a versão mais recente desse mecanismo. Para obter mais informações, consulte [Versões do ElastiCache for Memcached com suporte](#) ou [Versões do ElastiCache for Redis com suporte](#).

Migração online para o ElastiCache

Ao usar a migração on-line, você pode migrar seus dados do Redis de código-fonte aberto auto-hospedado no Amazon EC2 para o Amazon ElastiCache.

Note

A migração on-line não é compatível com caches ou clusters sem servidor do ElastiCache em execução no tipo de nó r6gd.

Visão geral

Migrar seus dados do Redis de código-fonte aberto em execução no Amazon EC2 para o Amazon ElastiCache exige uma implantação do Amazon ElastiCache existente ou recém-criada. A implantação deve ter uma configuração pronta para migração. Ela também deve estar alinhada à configuração desejada, incluindo atributos, como tipo de instância, número de fragmentos e número de réplicas.

A migração on-line foi desenvolvida para migrar os dados do Redis de código-fonte aberto hospedado no Amazon EC2 para o ElastiCache for Redis, e não entre clusters do ElastiCache for Redis.

Important

É altamente recomendável ler integralmente as seções a seguir antes de iniciar o processo de migração online.

A migração começa quando você chama o comando da AWS CLI ou a operação de API `StartMigration`. Para modo de cluster do Redis desabilitado, o processo de migração transforma o nó primário do cluster do ElastiCache para Redis em uma réplica do cluster do Redis primário. Para modo de cluster do Redis habilitado, o processo de migração transforma o nó primário de cada fragmento do ElastiCache em uma réplica do fragmento correspondente do cluster de origem que tem os mesmos slots.

Depois que as alterações no lado do cliente estiverem prontas, chame a operação de API `CompleteMigration`. Essa operação de API promove a implantação do ElastiCache para sua implantação primária do Redis com os nós principal e de réplica (conforme aplicável). Agora é possível redirecionar sua aplicação cliente para começar a gravar dados no ElastiCache. Durante toda a migração, é possível conferir o status da replicação executando o comando [redis-cli INFO](#) nos nós do Redis e nos nós primários do ElastiCache.

Etapas da migração

Os seguintes tópicos descrevem o processo de migração dos seus dados:

- [Preparação dos nós do Redis de origem e de 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](#)

Preparação dos nós do Redis de origem e de destino para a migração

É necessário garantir que todos os quatro pré-requisitos mencionados a seguir sejam cumpridos antes de iniciar a migração do console do ElastiCache, da API ou da CLI da AWS.

Como preparar os nós Redis de origem e de destino para a migração

1. Identifique a implantação do ElastiCache de destino e verifique se é possível migrar dados para ela.

Uma implantação do ElastiCache existente ou recém-criada deve atender aos seguintes requisitos para migração:

- Está usando o mecanismo do Redis versão 5.0.6 ou posterior.
 - Não tem a criptografia em trânsito nem a criptografia em repouso habilitada.
 - Ele tem o multi-AZ habilitado.
 - Tem memória suficiente disponível para comportar os dados do cluster do Redis. Para configurar as definições de memória reservada corretas, consulte [Gerenciamento da memória reservada](#).
 - No caso de desabilitado para cluster, você poderá migrar diretamente das versões 2.8.21 e posteriores para as versões 5.0.6 e posteriores do Redis se estiver usando a CLI ou as versões 5.0.6 e posteriores do Redis usando a CLI ou o console. No caso de habilitado para cluster, você poderá migrar diretamente de qualquer versão do Redis habilitada para cluster para as versões 5.0.6 e posteriores do Redis se estiver usando a CLI ou as versões 5.0.6 e posteriores do Redis usando a CLI ou o console.
 - O número de fragmentos na origem e no destino coincide.
 - Ele não faz parte de um datastore global.
 - Ele tem a hierarquização de dados desabilitada.
2. Verifique se as configurações do Redis e a implantação do ElastiCache para Redis de código-fonte aberto são compatíveis.

No mínimo, todos os itens a seguir na implantação do ElastiCache de destino devem ser compatíveis com a configuração do Redis para a replicação do Redis:

- O cluster do Redis não deve ter o Redis AUTH habilitado.
 - O config `protected-mode` do Redis deve estar definido como `no`.
 - Se você tiver a configuração `bind` no config do Redis, ela deve ser atualizada para permitir solicitações de nós do ElastiCache.
 - O número de bancos de dados lógicos deve ser o mesmo no nó do ElastiCache e no cluster do Redis. Esse valor é definido usando `databases` no config do Redis.
 - Os comandos do Redis que executam a modificação de dados não devem ser renomeados para permitir que a replicação dos dados seja bem-sucedida. Por exemplo, `sync`, `psync`, `info`, `config`, `command` e `cluster`.
 - Para replicar os dados do cluster do Redis para o ElastiCache, verifique se há CPU e memória suficientes para lidar com essa carga adicional. Essa carga vem do arquivo do RDB criado pelo cluster do Redis e transferido pela rede para o nó do ElastiCache.
 - Todas as instâncias do Redis no cluster de origem devem estar em execução na mesma porta.
3. Verifique se as instâncias do EC2 podem se conectar ao ElastiCache fazendo o seguinte:
- Garanta que cada endereço IP da instância seja privado.
 - Atribua ou crie a implantação do ElastiCache na mesma nuvem privada virtual (VPC) que o Redis na instância do EC2 (recomendado).
 - Se as VPCs forem diferentes, configure o emparelhamento de VPCs para permitir o acesso entre os nós. Para obter mais informações sobre o emparelhamento de VPCs, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#).
 - O grupo de segurança anexado às instâncias do Redis deve permitir o tráfego de entrada de nós do ElastiCache.
4. Verifique se a aplicação pode direcionar o tráfego para nós do ElastiCache após a conclusão da migração de dados. Para obter mais informações, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#).

Testar a migração de dados

Depois que todos os pré-requisitos forem atendidos, será possível iniciar a migração de dados usando o AWS Management Console, a API do ElastiCache ou a AWS CLI. O exemplo a seguir mostra como usar a CLI.

Teste a migração chamando o comando `test-migration` com os seguintes parâmetros:

- `--replication-group-id`: o ID do grupo de replicação para o qual os dados devem ser migrados.
- `--customer-node-endpoint-list`: lista de endpoints dos quais os dados devem ser migrados. A lista deve ter somente um elemento.

Veja a seguir um exemplo de como usar a CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

O ElastiCache validará a configuração da migração sem nenhuma migração real de dados.

Início da migração

Depois que todos os pré-requisitos forem preenchidos, será possível iniciar a migração de dados usando o AWS Management Console, a API do ElastiCache ou a AWS CLI. No modo de cluster habilitado, se a migração de slots for diferente, uma refragmentação será realizada antes da migração ao vivo. O exemplo a seguir mostra como usar a CLI.

Note

Recomendamos usar a API `TestMigration` para validar a configuração da migração. Mas isso é estritamente opcional.

Inicie a migração chamando o comando `start-migration` com os seguintes parâmetros:

- `--replication-group-id`: identificador do grupo de replicação de destino do ElastiCache.
- `--customer-node-endpoint-list`: uma lista de endpoints com endereços IP ou DNS e a porta em que o cluster de origem do Redis está sendo executado. A lista só pode ter um elemento para o modo de cluster desabilitado e o modo de cluster habilitado. Se você habilitou a replicação encadeada, o endpoint pode apontar para uma réplica em vez de um nó primário no cluster do Redis.

Veja a seguir um exemplo de como usar a CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Ao executar esse comando, o nó primário do ElastiCache (em cada fragmento) se configura para se tornar uma réplica da instância do Redis (no fragmento correspondente que tem os mesmos slots no redis habilitado para cluster). O status do cluster do ElastiCache é alterado para migrando e os dados começam a ser migrados da instância do Redis para o nó primário do ElastiCache. Dependendo do tamanho dos dados e da carga na instância do Redis, a migração pode levar um tempo para ser concluída. É possível conferir o andamento da migração executando o comando [redis-cli INFO](#) na instância do Redis e no nó primário do ElastiCache.

Após a replicação bem-sucedida, todas as gravações nas instâncias do Redis são propagadas para o cluster do ElastiCache. É possível usar os nós do ElastiCache para leituras. No entanto, não é possível gravar no cluster do ElastiCache. Se um nó primário do ElastiCache tiver outros nós de replica conectados a ele, esse nó de réplica continuará a replicar do nó primário do ElastiCache. Dessa maneira, todos os dados do cluster do Redis serão replicados para todos os nós no cluster do ElastiCache.

Se um nó primário do ElastiCache não puder se tornar uma réplica da instância do Redis, ele fará várias novas tentativas antes de finalmente se promover novamente a primário. O status do cluster do ElastiCache é, então, alterado para available (disponível) e é enviado um evento do grupo de replicação sobre a falha ao iniciar a migração. Para solucionar problemas como uma falha, verifique o seguinte:

- Observe o evento do grupo de replicação. Use qualquer informação específica do evento para corrigir a falha na migração.
- Se o evento não fornecer informações específicas, verifique se as diretrizes em [Preparação dos nós do Redis de origem e de destino para a migração](#) foram seguidas.
- Assegure que a configuração de roteamento para a VPC e as sub-redes permitam tráfego entre os nós do ElastiCache e as instâncias do Redis.
- Garanta que o grupo de segurança anexado às instâncias do Redis permita tráfego de entrada de nós do ElastiCache.
- Confira os logs das instâncias do Redis para receber 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 `master_link_status` do Redis está up no comando INFO dos nós primários do ElastiCache. Também é possível encontrar essa informação no console do ElastiCache. Selecione o cluster e, em CloudWatch metrics (Métricas do CloudWatch), observe o Primary Link Health Status (Status de integridade do link principal). Depois que o valor atinge 1, os dados estão sincronizados.
- É possível conferir se a réplica do ElastiCache está no estado on-line executando o comando INFO nas instâncias do Redis. Isso também fornece informações sobre o atraso da replicação.
- Verifique o buffer baixo de saída do cliente usando o comando [CLIENT LIST](#) nas instâncias do Redis.

Após a conclusão da migração de dados, eles serão sincronizados com qualquer gravação nova enviada ao nó primário pelo cluster do Redis.

Conclusão da migração de dados

Quando estiver pronto para transferir ao cluster do ElastiCache, use o comando `complete-migration` da CLI com os seguintes parâmetros:

- `--replication-group-id`: o identificador do grupo de replicação.
- `--force`: um valor que força a interrupção da migração sem garantir que os dados estejam em sincronia.

Veja um exemplo a seguir.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

À medida que você executa esse comando, o nó primário do ElastiCache (em cada fragmento) interrompe a replicação da instância do Redis e o promove a primário. Normalmente, essa promoção é concluída em alguns minutos. Para confirmar a promoção para primário, verifique o evento `Complete Migration successful for test-cluster`. Neste ponto, você pode direcionar sua aplicação para gravações e leituras do ElastiCache. O status do cluster do ElastiCache deve mudar de `migrating` (em migração) para `available` (disponível).

Se a promoção para primário falhar, o nó primário do ElastiCache continuará a replicar da instância do Redis. O cluster do ElastiCache continuará no status migrating (em migração) e será enviada uma mensagem de evento do grupo de replicação sobre a falha. Para resolver essa falha, observe o seguinte:

- Verifique o evento do grupo de replicação. Use informações específicas do evento para corrigir a falha.
- Você pode receber uma mensagem de evento sobre os dados não estarem em sincronia. Se isso ocorrer, verifique se o primário do ElastiCache pode replicar da instância do Redis e se os dois estão em sincronia. Se você ainda quiser interromper a migração, poderá executar o comando anterior com a opção `-force`.
- Você pode receber uma mensagem de evento se um dos nós do ElastiCache estiver passando por uma substituição. Você poderá tentar executar novamente a etapa de migração completa depois que a substituição for concluída.

Execução de migração de dados online usando o console

É possível usar o AWS Management Console para migrar os dados do cluster para o cluster do Redis.

Como executar a migração de dados online usando o console

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Crie um cluster do Redis ou escolha um cluster existente. Verifique se o cluster atende aos seguintes requisitos:
 - A versão do seu mecanismo do Redis deve ser pelo menos 5.0.6.
 - O cluster do Redis não deve ter o Redis AUTH habilitado.
 - O config `protected-mode` do Redis deve estar definido como `no`.
 - Se você tiver a configuração `bind` no config do Redis, ela deve ser atualizada para permitir solicitações de nós do ElastiCache.
 - O número de bancos de dados deve ser o mesmo entre o nó do ElastiCache e o cluster do Redis. Esse valor é definido usando `databases` no config do Redis.
 - Os comandos do Redis que executam a modificação de dados não devem ser renomeados para permitir que a replicação dos dados seja bem-sucedida.

- Para replicar os dados do cluster do Redis para o ElastiCache, verifique se há CPU e memória suficientes para lidar com essa carga adicional. Essa carga vem do arquivo do RDB criado pelo cluster do Redis e transferido pela rede para o nó do ElastiCache.
 - O cluster está no status available (disponível).
3. Com o cluster selecionado, escolha Migrate Data from Endpoint (Migrar dados do endpoint) para Actions (Ações).
 4. Na caixa de diálogo Migrar dados do endpoint, insira o endereço IP ou a porta onde o cluster do Redis está disponível.

 Important

O endereço IP deve ser exato. Se você inserir o endereço de maneira incorreta, a migração falhará.

5. Selecione Start Migration (Iniciar migração).

À medida que o cluster inicia a migração, ele muda para o status Modifying (Modificando) e, depois, para Migrating (Migrando).

6. Monitore o andamento da migração selecionando Events (Eventos) no painel de navegação.

A qualquer momento durante o processo de migração, é possível interromper a migração. Para fazer isso, escolha o cluster e selecione Stop Data Migration (Interromper a migração de dados) para Actions (Ações). Então, o cluster entrará no status Available (Disponível).

Se a migração for bem-sucedida, o cluster entrará no status Available (Disponível) e o log de eventos mostrará o seguinte:

```
Migration operation succeeded for replication group ElastiCacheClusterName.
```

Se a migração falhar, o cluster entrará no status Available (Disponível) e o log de eventos mostrará o seguinte:

```
Migration operation failed for replication group ElastiCacheClusterName.
```

Escolher regiões e zonas de disponibilidade

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

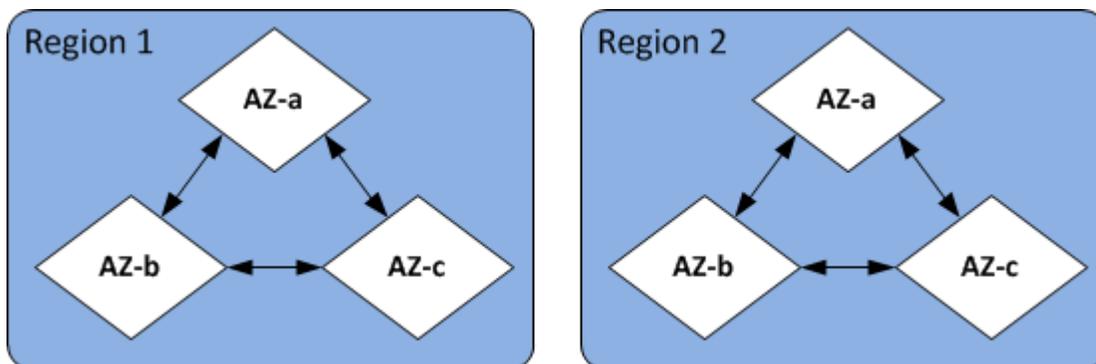
estão localizadas em diferentes locais físicos. Esses locais são categorizados por regiões e zonas de disponibilidade.

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

- [Localização dos seus nós](#)
- [Regiões e endpoints com suporte](#)
- [Uso de zonas locais com o ElastiCache](#)
- [Uso do Outposts](#)

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 várias AZs (recomendado), ElastiCache permite que você escolha a AZ para cada nó ou permita ElastiCache escolhê-las para você.

Ao localizar os nós em diferentes AZs, você elimina a chance de que uma falha, como uma queda de energia, em uma AZ faça com que todo o seu sistema falhe. O teste demonstrou que não há diferença de latência significativa entre localizar todos os nós em uma AZ ou espalhá-los em várias AZs.

Você pode especificar uma AZ para cada nó ao criar um cluster ou adicionar nós ao modificar um cluster existente. Para obter mais informações, consulte:

- [Criar um cluster](#)
- [Modificando um cluster ElastiCache](#)
- [Adição de nós a um 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 em diferentes AZs para obter tolerância a falhas. Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Escolher regiões e zonas de disponibilidade](#) 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	
Região Ásia-Pacífico (Sydney) ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo	
Região Europa (Frankfurt) eu-central-1	elasticache.eu-central-1.amazonaws.com	HTTPS	
Região Europa (Zurique) eu-central-2	elasticache.eu-central-2.amazonaws.com	HTTPS	
Região Europa (Estocolmo) eu-north-1	elasticache.eu-north-1.amazonaws.com	HTTPS	
Região Oriente Médio (Bahrein) me-south-1	elasticache.me-south-1.amazonaws.com	HTTPS	
Região do Oriente Médio (Emirados Árabes Unidos) me-central-1	elasticache.me-central-1.amazonaws.com	HTTPS	
Região Europa (Irlanda) eu-west-1	elasticache.eu-west-1.amazonaws.com	HTTPS	
Região Europa (Londres) eu-west-2	elasticache.eu-west-2.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo	
Região Europa (Paris) eu-west-3	elasticache.eu-west-3.amazonaws.com	HTTPS	
Região Europa (Milão) eu-south-1	elasticache.eu-south-1.amazonaws.com	HTTPS	
Região Europa (Espanha) eu-south-2	elasticache.eu-south-2.amazonaws.com	HTTPS	
Região América do Sul (São Paulo) sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS	
Região China (Pequim) cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS	
Região China (Ningxia) cn-northwest-1	elasticache.cn-northwest-1.amazonaws.com.cn	HTTPS	
Região Ásia-Pacífico (Hong Kong) ap-east-1	elasticache.ap-east-1.amazonaws.com	HTTPS	

Nome da região/região	Endpoint	Protocolo
Região África (Cidade do Cabo) af-south-1	elasticache.af-south-1.amazonaws.com	HTTPS
Região de Israel (Tel Aviv) il-central-1	elasticache.il-central-1.amazonaws.com	HTTPS
AWS GovCloud (Oeste dos EUA) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (Leste dos EUA) us-gov-east-1	elasticache.us-gov-east-1.amazonaws.com	HTTPS

Para obter informações sobre 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](#).

Uso de zonas locais com o ElastiCache

Uma zona local é uma extensão de uma região AWS que está geograficamente próxima de seus usuários. É possível estender qualquer nuvem privada virtual (VPC) de uma região principal da AWS para as zonas locais criando uma nova sub-rede e atribuindo-a a uma Local Zone. Quando você criar

uma sub-rede em uma zona local, sua VPC também será estendida para essa zona local. A sub-rede na zona local funciona da mesma forma que outras sub-redes na VPC.

Local Zones (Zonas locais) permitem que você coloque recursos como um cluster do ElastiCache em vários locais próximos aos seus usuários.

Quando criar um cluster do ElastiCache, você poderá escolher uma sub-rede em uma zona local. As zonas locais têm suas próprias conexões com a Internet e suporte no 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 região da AWS seguido por um identificador que indica o local, por exemplo, `us-west-2-lax-1a`.

Neste momento, zonas locais disponíveis são `us-west-2-lax-1a` e `us-west-2-lax-1b`.

As seguintes limitações se aplicam ao ElastiCache para zonas locais:

- Não há suporte para os armazenamentos de dados globais.
- Não há suporte para migração online.
- Há suporte de zonas locais para os tipos de nó a seguir no momento:
 - Geração atual:

Tipos de nó 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. Habilite a zona local no console do Amazon EC2.

Para obter mais informações, consulte [Habilitação de zonas locais](#) no Guia do usuário do Amazon EC2.

2. Crie uma sub-rede na zona local.

Para obter mais informações, consulte [Criar uma sub-rede na VPC](#) no Guia do usuário da Amazon VPC.

3. Criar um grupo de sub-redes do ElastiCache na zona local.

Ao criar um grupo de sub-redes do ElastiCache, escolha o grupo de zonas de disponibilidade para a zona local.

Para obter mais informações, consulte [Criação de um grupo de sub-redes](#) no Guia do usuário do ElastiCache.

4. Criar um cluster do ElastiCache for Redis que use a sub-rede do ElastiCache na zona local.

Para obter mais informações, consulte um dos tópicos a seguir:

- [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)

Uso do Outposts

O Outposts da AWS é um serviço totalmente gerenciado que estende a infraestrutura, os serviços, as APIs e as ferramentas da AWS no local do cliente. Ao fornecer acesso local à infraestrutura gerenciada pela AWS, o Outposts da AWS permite que os clientes criem e executem aplicações on-premises usando as mesmas interfaces de programação que nas regiões da AWS, ao mesmo tempo que usam recursos locais de computação e armazenamento para menor latência e necessidades de processamento de dados locais. Um Outpost é um grupo de capacidade de computação e armazenamento da AWS implantado em um local do cliente. A AWS opera, monitora e gerencia essa capacidade como parte de uma região da AWS. Você pode criar sub-redes no Outpost e especificá-las ao criar recursos da AWS como clusters do ElastiCache.

Note

Nesta versão, as seguintes limitações se aplicam:

- O ElastiCache para Outposts oferece suporte somente às famílias de nós M5 e R5.
- Não há suporte para migração ao vivo.
- Multi-AZ (não há suporte para replicação cruzada do Outpost).
- Não há suporte para snapshots locais.

- O ElastiCache para Outposts não tem suporte nas seguintes regiões: cn-north-1, cn-northwest-1 e ap-northeast-3.

Uso de Outposts com o console do Redis

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
3. Em Cluster engine (Mecanismo de cluster), selecione Redis.
4. Em Local, selecione On-Premises: criar suas instâncias do ElastiCache no AWS Outposts.

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. Em Redis settings (Configurações do Redis):
 - a. Name (Nome): insira um nome para o cluster do Redis.
 - b. Description (Descrição): insira uma descrição para o cluster do Redis.
 - c. Engine version compatibility (Compatibilidade da versão do mecanismo): a versão do mecanismo é baseada na região do Outpost da AWS
 - d. Port (Porta): aceite a porta padrão, 6379. Se você tiver um motivo para 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. O assistente de portabilidade para .NET para Outposts oferece suporte somente a 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 Advanced Redis settings (Configurações avançadas do Redis):

a. Subnet Group (Grupo de sub-redes): na lista, selecione Create new (Criar novo).

- Name (Nome): insira um nome para o grupo de sub-redes.
- Description (Descrição): insira uma descrição para o grupo de sub-redes.
- VPC ID (ID da VPC): o ID da VPC deve corresponder à VPC do Outpost. Se você selecionar uma VPC que não tenha IDs de sub-rede 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 IDs de sub-rede disponíveis, você precisará criá-los. Para obter mais informações, consulte [Criar uma sub-rede](#).

b. Escolha Create (Criar).

Visualização de detalhes do cluster do Outpost

Na página de lista do Redis, selecione um cluster que pertença a um Outpost da AWS e observe o seguinte ao visualizar os Cluster details (Detalhes do cluster):

- Availability Zone (Zona de disponibilidade): isso representará o Outpost, usando um ARN (Nome de recurso da Amazon) e o número do recurso da AWS.
- Outpost name (Nome do Outpost): o nome do Outpost da AWS.

Uso de Outposts com a CLI da AWS

Você pode usar a AWS Command Line Interface (AWS CLI) para controlar vários serviços da AWS via linha de comando e automatizá-los usando scripts. Você pode usar a CLI da AWS para operações ad hoc (únicas).

Download e configuração da AWS CLI

A AWS CLI é executada no Windows, no macOS ou no Linux. Use o procedimento a seguir para fazer download e configurá-la.

Como fazer download, instalar e configurar a CLI

1. Baixe a CLI da AWS na página da Web [Interface de linha de comando da AWS](#).
2. Siga as instruções em [Instalação da CLI da AWS](#) e [Configuração da CLI da AWS](#) no Guia do usuário da AWS Command Line Interface.

Uso da CLI da AWS com Outposts

Use a seguinte operação da CLI para criar um cluster de cache que use Outposts:

- [create-cache-cluster](#): usando esta operação, o parâmetro `outpost-mode` 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 de mecanismos e atualização](#)
- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)
- [Gerenciamento do cluster autoprojeto](#)
- [Dimensionamento ElastiCache para Redis](#)
- [Conceitos básicos do JSON no ElastiCache para Redis](#)
- [Marcação de recursos do ElastiCache](#)
- [Usar a Lente do Well-Architected para o Amazon ElastiCache](#)
- [Etapas comuns de solução de problemas e melhores práticas](#)
- [Etapas adicionais de solução de](#)

Snapshots e restauração

Os caches do Amazon ElastiCache nos quais o Redis esteja em execução podem fazer backup dos dados criando um snapshot. É possível usar o backup para restaurar um cache ou propagar dados para um novo cache. O backup consiste nos metadados do cache com todos os dados no cache. Todos os backups são gravados no Amazon Simple Storage Service (Amazon S3), que fornece armazenamento durável. A qualquer momento, você pode restaurar os dados criando um novo cache do Redis e o preenchendo com dados de um backup. Com o ElastiCache, é possível gerenciar backups usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) e a API do ElastiCache.

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 autoprojeto](#)
- [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](#)
- [Propagação de um novo cluster autoprojeto com um backup criado externamente](#)

Restrições do backup

Considere as seguintes restrições ao planejar ou fazer backups:

- No momento, o backup e a restauração só são compatíveis com caches no Redis.
- Para clusters do Redis (modo cluster desabilitado), não há suporte para backup e restauração em nós `cache.t1.micro`. Todos os outros tipos de nó de cache têm suporte.
- Para clusters do Redis (modo cluster habilitado), o backup e a restauração têm suporte para todos os tipos de nó.
- Durante qualquer período contíguo de 24 horas, você não pode criar mais de 20 backups manuais por nó no cluster.
- O Redis (modo cluster habilitado) apenas oferece suporte para backups no nível do cluster (para a API ou CLI, no nível do grupo de replicação). O Redis (modo cluster habilitado) não oferece suporte à execução de backups no nível do fragmento (para a API ou CLI, no nível do grupo de nós).
- Durante o processo de backup, não é possível executar outra operação da API ou CLI no cluster.
- Se estiver usando clusters com classificação de dados em níveis, você 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.

Estas são as diretrizes a fim de melhorar o desempenho do backup para clusters autoprotetidos.

- Defina o parâmetro `reserved-memory-percent`: para evitar o excesso de paginação, recomendamos definir o parâmetro `reserved-memory-percent`. Esse parâmetro impede que o Redis consuma 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 os parâmetros `reserved-memory` e `reserved-memory-percent`, consulte [Gerenciamento da memória reservada](#).
- Crie backups a partir de uma réplica de leitura: se você estiver executando o Redis 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 um BGSAVE, recomendamos que você crie backups de uma das réplicas de leitura. Enquanto o backup está sendo criado da réplica, o nó primário permanece inalterado pelos requisitos de recursos de BGSAVE. O nó primário pode continuar a atender solicitações sem ficar lento.

Para fazer isso, consulte [Criação de um backup manual \(console\)](#), e no campo Cluster Name (Nome do cluster) na janela Create Backup (Criar backup), escolha uma réplica em vez do nó primário padrão.

Se você excluir um grupo de replicação e solicitar um backup final, o ElastiCache sempre usará o backup do nó primário. Isso garante que você capture os dados mais recentes do Redis antes que o grupo de replicação seja excluído.

Programação de backups automáticos

É possível habilitar backups automáticos para qualquer cache do ElastiCache para Redis. Quando os backups automáticos estão habilitados, o ElastiCache cria diariamente um backup do cluster. 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 quando o ElastiCache começa a criar um backup. É possível definir a janela de backup para qualquer momento quando mais prático. Se uma janela de backup não for especificada, o 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.

Você pode habilitar ou desabilitar os backups automáticos ao criar um novo cache ou atualizar um cache do Redis existente, usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache marcando a caixa Habilitar backups automáticos na seção Configurações avançadas do Redis.

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, a AWS CLI, ou a API do ElastiCache.

Criação de um backup manual (console)

Para criar um backup de um cache (console)

1. Faça login no AWS Management Console e abra o console Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação, escolha Caches do Redis.
3. Escolha a caixa à esquerda do nome do cluster Redis 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 a AWS CLI

Para criar um backup manual de um cache usando a AWS CLI, use a operação `create-serverless-snapshot` da AWS CLI 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 autoprotetado com a AWS CLI

Para criar um backup manual de um cluster autoprotetado usando a AWS CLI, use a operação `create-snapshot` da AWS CLI com os seguintes parâmetros:

- `--cache-cluster-id`
 - Se o cluster do backup não tiver nós de réplica, `--cache-cluster-id` será o nome dele, 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 poderá ser *mycluster-002*.

Use este parâmetro somente ao fazer backup de um cluster do Redis (modo cluster desabilitado).

- `--replication-group-id`: nome do cluster do Redis (modo cluster habilitado) (CLI/API: um grupo de replicação) para usar como fonte para o backup. Use este parâmetro ao fazer backup de um cluster do Redis (modo cluster habilitado).
- `--snapshot-name` - Nome do snapshot a ser criado.

As restrições de nomenclatura de cluster são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

Exemplo 1: backup de um cluster do Redis (modo cluster desabilitado) que não tem nós de réplicas

A seguinte operação da AWS CLI cria o backup `bkup-20150515` do cluster do Redis (modo cluster desabilitado) `myNonClusteredRedis` que não possui réplicas de leitura.

Para Linux, macOS ou Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis \  
  --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis ^  
  --snapshot-name bkup-20150515
```

Exemplo 2: backup de um cluster do Redis (modo cluster desabilitado) com nós de réplicas

A seguinte operação da AWS CLI cria o backup `bkup-20150515` do cluster do Redis (modo cluster desabilitado) `myNonClusteredRedis`. Esse backup tem uma ou mais réplicas de leitura.

Para Linux, macOS ou Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis-001 \  
  --snapshot-name bkup-20150515
```

Para Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis-001 ^  
  --snapshot-name bkup-20150515
```

Saída do exemplo: backup de um cluster do Redis (modo cluster desabilitado) com nós de réplicas

A saída da operação é semelhante à seguinte.

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x",  
    "VpcId": "vpc-91280df6",  
    "CacheClusterId": "myNonClusteredRedis-001",  
    "SnapshotRetentionLimit": 0,  
    "NumCacheNodes": 1,  
    "SnapshotName": "bkup-20150515",  
    "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",  
    "AutoMinorVersionUpgrade": true,  
    "PreferredAvailabilityZone": "us-east-1c",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "SnapshotWindow": "08:30-09:30",  
    "EngineVersion": "6.0",  
    "NodeSnapshots": [  
      {  
        "CacheSize": "",  
        "CacheNodeId": "0001",  
        "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"  
      }  
    ],  
    "CacheSubnetGroupName": "default",  
    "Port": 6379,  
    "PreferredMaintenanceWindow": "wed:07:30-wed:08:30",
```

```

    "CacheNodeType": "cache.m3.2xlarge",
    "DataTiering": "disabled"
  }
}

```

Exemplo 3: backup de um cluster do Redis (modo cluster habilitado)

A seguinte operação da AWS CLI cria o backup `bkup-20150515` do cluster do Redis (modo cluster habilitado) `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

```

Saída do exemplo: backup de um cluster do Redis (modo cluster habilitado)

A saída dessa operação é semelhante à seguinte.

```

{
  "Snapshot": {
    "Engine": "redis",
    "CacheParameterGroupName": "default.redis6.x.cluster.on",
    "VpcId": "vpc-91280df6",
    "NodeSnapshots": [
      {
        "CacheSize": "",
        "NodeGroupId": "0001"
      },
      {
        "CacheSize": "",
        "NodeGroupId": "0002"
      }
    ],
  },
}

```

```
    "NumNodeGroups": 2,  
    "SnapshotName": "bkup-20150515",  
    "ReplicationGroupId": "myClusteredRedis",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "enabled",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "SnapshotWindow": "10:00-11:00",  
    "EngineVersion": "6.0",  
    "CacheSubnetGroupName": "default",  
    "ReplicationGroupDescription": "2 shards 2 nodes each",  
    "Port": 6379,  
    "PreferredMaintenanceWindow": "sat:03:30-sat:04:30",  
    "CacheNodeType": "cache.r3.large",  
    "DataTiering": "disabled"  
  }  
}
```

Tópicos relacionados 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 console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Criação de um backup final (console)

Você pode criar um backup final ao excluir um cache do Redis usando o console do ElastiCache.

Para criar um backup final ao excluir um cache do Redis, na caixa de diálogo, escolha Sim, em Criar backup e dê um nome ao backup.

Tópicos relacionados da

- [Usar a 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 do Redis usando a AWS CLI.

Tópicos

- [Durante a exclusão de um cache sem servidor do Redis](#)
- [Ao excluir um cluster do Redis sem réplicas de leitura](#)
- [Ao excluir um cluster Redis com réplicas de leitura](#)

Durante a exclusão de um cache sem servidor do Redis

Para criar um backup final, use a operação `delete-serverless-cache` AWS CLI 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 comando da AWS CLI.

Ao excluir um cluster do Redis sem réplicas de leitura

Para criar um backup final, use a operação `delete-cache-cluster` AWS CLI com os seguintes parâmetros.

- `--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 comando da AWS CLI.

Ao excluir um cluster Redis com réplicas de leitura

Para criar um backup final ao excluir um grupo de replicação, use a operação `delete-replication-group` AWS CLI, com os seguintes parâmetros:

- `--replication-group-id` – Nome do grupo de replicação que está sendo excluído.
- `--final-snapshot-identifier` – Nome do backup final.

O código a seguir usa o backup final `bkup-20150515-final` ao excluir o grupo de replicação `myReplGroup`.

Para Linux, macOS ou Unix:

```
aws elasticache delete-replication-group \  
    --replication-group-id myReplGroup \  
    --final-snapshot-identifier bkup-20150515-final
```

Para Windows:

```
aws elasticache delete-replication-group ^  
    --replication-group-id myReplGroup ^  
    --final-snapshot-identifier bkup-20150515-final
```

Para obter mais informações, consulte [delete-replication-group](#) na Referência de comandos da AWS CLI.

Descrição de backups

Os procedimentos a seguir mostram como exibir uma lista dos seus backups. Se desejar, você também pode visualizar os detalhes de um backup específico.

Descrição de backups (console)

Para exibir backups usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Para ver os detalhes de um backup específico, escolha a caixa à esquerda do nome do backup.

Descrição dos backups sem servidor (AWS CLI)

Para exibir uma lista de backups sem servidor e, como opção, detalhes sobre um backup específico, use a operação `describe-serverless-cache-snapshots` da CLI.

Exemplos

A seguinte operação usa o parâmetro `--max-records` para listar até 20 backups associados à sua conta. Omitir o parâmetro `--max-records` lista até 50 backups.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

A operação a seguir usa o parâmetro `--serverless-cache-name` para só listar os backups associados ao cache `my-cache`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

A operação a seguir usa o parâmetro `--serverless-cache-snapshot-name` para exibir os detalhes do backup `my-backup`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

Para obter mais informações, consulte [describe-serverless-cache-snapshots](#) na Referência de comando da AWS CLI.

Descrição dos backups de cluster autoprojeto (AWS CLI)

Para exibir uma lista de backups autoprojeto e, como opção, detalhes sobre um backup específico, use a operação `describe-snapshots` da CLI.

Exemplos

A seguinte operação usa o parâmetro `--max-records` para listar até 20 backups associados à sua conta. Omitir o parâmetro `--max-records` lista até 50 backups.

```
aws elasticache describe-snapshots --max-records 20
```

A operação a seguir usa o parâmetro `--cache-cluster-id` para listar apenas os backups associados ao cluster `my-cluster`.

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

A operação a seguir usa o parâmetro `--snapshot-name` para exibir os detalhes do backup `my-backup`.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Para obter mais informações, consulte [describe-snapshots](#) na Referência de comandos da 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 poder acessá-lo de fora do 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 console do ElastiCache 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 autoprotetado (AWS CLI)

Para copiar um backup de um cluster autoprotetado, 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 ElastiCache backup para um bucket do Amazon Simple Storage Service (Amazon S3), que dá acesso externo a ele. ElastiCache Você pode exportar um backup usando o ElastiCache console AWS CLI, o ou a ElastiCache API.

Exportar um backup pode ser útil se você precisar iniciar um cluster em outra AWS região. Você pode exportar seus dados em uma AWS região, copiar o arquivo.rdb para a nova AWS região e, em seguida, usar esse arquivo.rdb para semear o novo cluster em vez de esperar que o novo cluster seja preenchido por meio do uso. Para obter informações sobre como criar um novo cluster, consulte [Propagação de um novo cluster autoprojeto com um backup criado externamente](#). Outra razão pela qual você pode querer exportar os dados do seu cluster é para usar o arquivo .rdb para processamento offline.

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 ter mais informações, consulte [Classificação de dados em níveis](#).

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.

Etapa 1: Crie 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 AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.](https://console.aws.amazon.com/s3/)
2. Escolha Criar bucket.
3. Em Create a Bucket - Select a Bucket Name and Region, faça o seguinte:
 - a. Em Bucket Name (Nome do bucket), digite um nome para o bucket do Amazon S3.

O nome do bucket do Amazon S3 deve estar em conformidade com o DNS. Caso contrário, não ElastiCache conseguirá acessar seu arquivo de backup. As regras para a conformidade de DNS são:

- Os nomes devem ter no mínimo 3 e no máximo 63 caracteres de extensão.
 - Os nomes devem ser uma série de um ou mais rótulos separados por um ponto (.) em que cada rótulo:
 - Começa com uma letra minúscula ou um número.
 - Termina com uma letra minúscula ou um número.
 - Contém somente letras minúsculas, números e traços.
 - Os nomes não podem ser formatado como um endereço IP (por exemplo, 192.0.2.0).
- b. Na lista de regiões, escolha uma AWS região para seu bucket do Amazon S3. Essa AWS região deve ser a mesma AWS região do ElastiCache backup que você deseja exportar.
 - c. Escolha Criar.

Para obter mais informações sobre como criar um bucket do Amazon S3, consulte [Criação de um bucket](#), no Guia do usuário do Amazon Simple Storage Service.

Etapa 2: Conceder ElastiCache acesso ao seu bucket do 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, recomendamos que você configure políticas do IAM para impedir o

acesso não autorizado a esse bucket do Amazon S3. Para obter mais informações, consulte [Gerenciamento de acesso](#) no Guia do usuário do Amazon S3.

Para criar as permissões adequadas em um bucket do Amazon S3, siga as etapas descritas abaixo.

Para conceder ElastiCache acesso a um bucket do S3

1. [Faça login AWS Management Console e abra o console do Amazon S3 em https://console.aws.amazon.com/s3/](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 [Etapa 1: Crie um bucket do Amazon S3](#).
3. Selecione a guia Permissions (Permissões) e, em Permissions (Permissões), escolha Access control list (ACL) (Lista de controle de acesso (ACL)) e, depois, Edit (Editar).
4. Adicionar ID canônico do beneficiário
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353 com as seguintes opções:
 - Objetos: listar, gravar
 - ACL de bucket: ler, gravar

 Note

- Para a GovCloud região PDT, o ID canônico é.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Para a GovCloud região OSU, o ID canônico é.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Selecione Salvar.

Etapa 3: exportar um ElastiCache backup

Agora você criou seu bucket do S3 e concedeu ElastiCache permissões para acessá-lo. Em seguida, você pode usar o ElastiCache console, a AWS CLI ou a ElastiCache API para exportar seu snapshot para ele. Os seguintes exemplos pressupõem que a identidade do IAM do chamador tenha as seguintes permissões adicionais do IAM específicas 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 [Etapa 1: Crie 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 ter mais informações, consulte [Etapa 2: Conceder ElastiCache acesso ao seu bucket do 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 [Etapa 2: Conceder ElastiCache acesso ao seu bucket do Amazon S3](#) a fim de adicionar as permissões especificadas e tente exportar o backup novamente.

- ElastiCache não recebeu permissões de LEITURA %s no S3 Bucket.

Solução: adicione permissões de Leitura no bucket.

- ElastiCache não recebeu permissões de GRAVAÇÃO %s no S3 Bucket.

Solução: adicione permissões de Gravação no bucket.

- ElastiCache não recebeu as permissões READ_ACP %s no S3 Bucket.

Solução: adicione permissão de acesso de Leitura no bucket.

Se você quiser copiar seu backup para outra AWS região, use o Amazon S3 para copiá-lo. Para obter mais informações, consulte [Cópia de objetos](#) no Guia do usuário do Amazon Simple Storage Service.

Exportação de um ElastiCache backup sem servidor ()AWS CLI

Exportação de um backup de um cache sem servidor

Exporte o backup para um bucket do Amazon S3 usando a operação `export-serverless-cache-snapshot` da CLI com os seguintes parâmetros:

Parâmetros

- `--serverless-cache-snapshot-name` - Nome do backup a ser copiado.

- `--s3-bucket-name`: nome do bucket do Amazon S3 no qual você deseja exportar o backup. Uma cópia do backup é feita no bucket especificado.
 - `--s3-bucket-name` Deve ser um bucket do Amazon S3 na AWS região do backup com as seguintes permissões para que o processo de exportação seja bem-sucedido.
 - Acesso ao objeto: Ler e Escrever.
 - Permissões de acesso: Ler.

A operação a seguir copia um backup para `my-s3-bucket`.

Para Linux, macOS ou Unix:

```
aws elasticache export-serverless-cache-snapshot \  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \  
  --s3-bucket-name my-s3-bucket
```

Para Windows:

```
aws elasticache export-serverless-cache-snapshot ^  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^  
  --s3-bucket-name my-s3-bucket
```

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 operação `copy-snapshot` da CLI com os seguintes parâmetros:

Parâmetros

- `--source-snapshot-name` - Nome do backup a ser copiado.
- `--target-snapshot-name` - Nome da cópia do backup.

O nome deve ter entre 1 e 1.000 caracteres e pode ser codificado em UTF-8.

ElastiCache adiciona um identificador de instância e `.rdb` ao valor inserido aqui. Por exemplo, se você inserir `my-exported-backup`, ElastiCache criará `my-exported-backup-0001.rdb`.

- `--target-bucket`: nome do bucket do Amazon S3 no qual você deseja exportar o backup. Uma cópia do backup é feita no bucket especificado.

--target-bucket Deve ser um bucket do Amazon S3 na AWS região do backup com as seguintes permissões para que o processo de exportação seja bem-sucedido.

- Acesso ao objeto: Ler e Escrever.
- Permissões de acesso: Ler.

Para ter mais informações, consulte [Etapa 2: Conceder ElastiCache acesso ao seu bucket do 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

É possível restaurar um backup existente em um novo cache sem servidor ou em um cluster autoprotetido.

Restauração de um backup para um cache sem servidor (console)

Note

O ElastiCache Sem Servidor dá suporte a arquivos RDB compatíveis com as 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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup 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 autoprotetido (console)

Para restaurar um backup para um cluster autoprotetido (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Backups.
3. Na lista de backups, escolha a caixa à esquerda do nome do backup do qual você deseja restaurar.
4. Escolha 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 autoprotetado e importe dados do backup.

Restauração de um backup para um cache sem servidor (AWS CLI)

Note

O ElastiCache Sem Servidor dá suporte a arquivos RDB compatíveis com as 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 exemplo da AWS CLI 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 autoprotetado (AWS CLI)

Para restaurar um backup para um cluster autoprotetado (AWS CLI)

É possível restaurar um backup do Redis (modo cluster desabilitado) de duas maneiras.

- Você pode restaurar para um cluster do Redis de nó único (modo cluster desabilitado) usando a operação `create-cache-cluster` da AWS CLI.
- Ou é possível restaurar para um cluster do Redis com réplicas de leitura (um grupo de replicações). Para isso, é possível usar o Redis (modo cluster desabilitado) ou o Redis (modo cluster habilitado) com a operação `create-replication-group` da AWS CLI. Neste caso, propague

a restauração com um arquivo .rdb do Redis. Para obter mais informações sobre como propagar um novo cluster autoprotetido, consulte [Propagação de um novo cluster autoprotetido 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.

O ElastiCache fornece uma operação de API de exclusão que permite excluir um backup a qualquer momento, independentemente de o backup ter sido criado automaticamente 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 console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Exclusão de um backup (console)

O procedimento a seguir exclui um backup usando o console do ElastiCache.

Para excluir um backup

1. Faça login no AWS Management Console e abra o console do ElastiCache 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 Delete (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 operação `delete-snapshot` da AWS CLI 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 comando da AWS CLI.

Exclusão de um backup de cluster autoprotetido (AWS CLI)

Use a operação `delete-snapshot` da AWS CLI com o parâmetro a seguir para excluir um backup de cluster autoprotetido.

- `--snapshot-name` - Nome do backup a ser excluído.

O código a seguir exclui o backup `myBackup`.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Para obter mais informações, consulte [delete-snapshot](#) na Referência de comandos da AWS CLI.

Marcação de backups

Você pode atribuir os próprios metadados a cada backup na forma de tags. As tags permitem categorizar seus backups de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando você tem muitos recursos do mesmo tipo. É possível identificar rapidamente um recurso específico baseado nas tags que você atribuiu a ele. Para obter mais informações, consulte [Recursos que podem ser marcados](#).

Tags de alocação de custos são um meio de rastrear seus custos em vários serviços da AWS, agrupando suas despesas em faturas por valores de tag. Para saber mais sobre alocação de custos, consulte [Usar tags de alocação de custos](#).

Usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache, você pode adicionar, listar, modificar, remover ou copiar tags de alocação de custos em seus backups. Para obter mais informações, consulte [Monitoramento de custos com tags de alocação de custos](#).

Propagação de um novo cluster autoprojetoado com um backup criado externamente

Ao criar um novo cluster autoprojetoado do Redis, você pode propagá-lo com dados de um arquivo de backup .rdb do Redis. A propagação do cluster será útil se você gerenciar atualmente uma instância do Redis fora do ElastiCache e quiser preencher o novo cluster autoprojetoado do ElastiCache para Redis com os dados do Redis existentes.

Para propagar um novo cluster autoprojetoado do Redis de um backup do Redis criado dentro do Amazon ElastiCache, consulte [Restauração de um backup para um novo cache](#).

Ao usar um arquivo .rdb do Redis para propagar um novo cluster autoprojetoado do Redis, você pode fazer o seguinte:

- Atualização de um cluster não particionado para um cluster autoprojetoado do Redis (modo cluster habilitado) executando o Redis versão 3.2.4.
- Especifique um número de fragmentos (chamados de grupos de nós na API ou na CLI) 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 dimensionar 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 Redis. Para obter mais informações, consulte [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#).
- Distribua suas chaves nos slots do novo cluster do Redis (modo cluster habilitado) de forma diferente do que no cluster que foi usado para criar o arquivo de backup.

Note

Não é possível propagar um cluster do Redis (modo cluster desabilitado) a partir de um arquivo .rdb criado a partir de um cluster do Redis (modo cluster habilitado).

⚠ Important

- É necessário garantir que seus dados de backup do Redis não excedam os recursos do nó. Por exemplo, você não pode fazer upload de um arquivo .rdb com 5 GB de dados do Redis em um nó cache.m3.medium que tenha 2,9 GB de memória.

Se o backup for muito grande, o cluster resultante terá um status de `restore-failed`. Se isso acontecer, você deverá excluir o cluster e começar de novo.

Para obter uma lista completa dos tipos e das especificações de nós, consulte [Parâmetros específicos de node-type do Redis](#) e [Recursos do produto e detalhes do Amazon ElastiCache](#).

- Só é possível criptografar um arquivo .rdb do Redis com a criptografia no lado do servidor do Amazon S3 (SSE-S3). Para obter mais informações, consulte [Proteger dados usando a criptografia no lado do servidor](#).

A seguir, você pode encontrar tópicos que demonstram passo a passo como migrar seu cluster do Redis de fora do ElastiCache para Redis no ElastiCache para Redis.

Migração para o ElastiCache para Redis

- [Etapa 1: criar um backup do Redis](#)
- [Etapa 2: criar um bucket e uma pasta no Amazon S3](#)
- [Etapa 3: carregar seu backup no Amazon S3](#)
- [Etapa 4: conceder acesso de leitura do ElastiCache para o arquivo .rdb](#)

Etapa 1: criar um backup do Redis

Para criar o backup do Redis para propagar sua instância do ElastiCache para Redis

1. Conecte-se à sua instância do Redis existente.
2. Execute a operação BGSAVE ou SAVE do Redis para criar um backup. Observe onde seu arquivo .rdb está localizado.

BGSAVE é assíncrono e não bloqueia outros clientes durante o processamento. Para obter mais informações, consulte [BGSAVE](#) no site do Redis.

SAVE é síncrono e bloqueia outros processos até terminar. Para obter mais informações, consulte [SAVE](#) no site do Redis.

Para obter informações adicionais sobre como criar um backup, consulte o tópico sobre [Persistência do Redis](#) no site do Redis.

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 bucket do Amazon S3 deve estar em conformidade com o DNS. Caso contrário, o ElastiCache não poderá acessar seu arquivo de backup. As regras para a conformidade de DNS são:

- Os nomes devem ter no mínimo 3 e no máximo 63 caracteres de extensão.
- Os nomes devem ser uma série de um ou mais rótulos separados por um ponto (.) em que cada rótulo:
 - Começa com uma letra minúscula ou um número.
 - Termina com uma letra minúscula ou um número.
 - Contém somente letras minúsculas, números e traços.
- Os nomes não podem ser formatado como um endereço IP (por exemplo, 192.0.2.0).

Você deve criar seu bucket do Amazon S3 na mesma região da AWS que seu novo cluster do ElastiCache para Redis. Essa abordagem garante a maior velocidade possível de transferência de dados quando o ElastiCache lê seu arquivo .rdb do Amazon S3.

Note

Para manter seus dados da forma mais segura possível, restrinja ao máximo as permissões em seu bucket do Amazon S3. Ao mesmo tempo, as permissões ainda precisam permitir que o bucket e seu conteúdo seja usado para propagar o novo cluster do Redis.

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 do Redis](#). 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 leitura do ElastiCache para o arquivo .rdb

Agora, conceda acesso de leitura do ElastiCache para o seu arquivo de backup .rdb. Você concede acesso do ElastiCache ao arquivo de backup de uma maneira diferente, dependendo de o bucket estar em uma região padrão da AWS ou em uma região opcional da AWS.

Regiões da AWS adicionadas antes de 20 de março de 2019 são habilitadas por padrão. É possível começar a trabalhar nessas regiões da AWS 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 região da AWS:

- Para uma região padrão, use o procedimento em [Conceder acesso de leitura do ElastiCache ao arquivo .rdb em uma região padrão](#).
- Para uma região opcional, use o procedimento em [Conceder acesso de leitura do ElastiCache ao arquivo .rdb em uma região de opção](#).

Conceder acesso de leitura do ElastiCache ao arquivo .rdb em uma região padrão

Regiões da AWS adicionadas antes de 20 de março de 2019 são habilitadas por padrão. É possível começar a trabalhar nessas regiões da AWS 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.

Como conceder acesso de leitura do ElastiCache ao arquivo de backup em uma região da AWS habilitada 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-s3-readonly` ou um dos IDs canônicos na lista a seguir não estiver listado como usuário, faça o seguinte:
 - a. Em Access for other AWS accounts (Acesso para outras contas da), escolha Add grantee (Adicionar beneficiário).
 - b. Na caixa, adicione o ID canônico da região da AWS, conforme mostrado a seguir:
 - Região GovCloud (Oeste dos EUA) da AWS:

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

 Important

O backup deve estar localizado em um bucket do S3 na região AWS GovCloud (US) para poder ser transferido por download para um cluster do Redis nessa região AWS GovCloud (US).

- Regiões da AWS habilitadas por padrão:

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

- c. Defina as permissões no bucket escolhendo Yes (Sim) para o seguinte:
 - Objeto de lista/gravação

- Permissões ACL de objetos de leitura/gravação
- d. Escolha Save (Salvar).
7. Escolha Overview (Visão geral) e escolha Download (Fazer download).

Conceder acesso de leitura do ElastiCache ao arquivo .rdb em uma região de opção

Regiões da AWS adicionadas antes de 20 de março de 2019 são habilitadas por padrão. É possível começar a trabalhar nessas regiões da AWS 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 leitura do ElastiCache para o seu arquivo de backup .rdb.

Para conceder acesso de leitura para o ElastiCache 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 Permissions (Permissões), escolha Bucket policy (Política de bucket) e, em seguida, Edit (Editar).
7. Atualize a política para conceder ao ElastiCache as permissões necessárias para executar 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 Save changes (Salvar alterações).

Etapa 5: propagar o cluster do ElastiCache com os dados do arquivo .rdb

Agora você está pronto para criar um cluster do ElastiCache e propagá-lo com dados do arquivo .rdb. Para criar o cluster, siga as instruções em [Criar um cluster](#) ou [Criação de um grupo de replicação do Redis do princípio](#). Certifique-se de escolher o Redis como seu mecanismo de cluster.

O método que você usa para dizer ao ElastiCache onde encontrar o backup do Redis que você carregou no Amazon S3 depende do método usado para criar o cluster:

Crie o cluster ou grupo de replicação do ElastiCache para Redis usando os dados do arquivo .rdb

- Uso do console do ElastiCache

Ao selecionar Cluster settings (Configurações de cluster), escolha Restore from backups (Restaurar de backups) como seu método de criação de cluster e escolha Other backups (Outros backups) como sua Source (Origem) na seção Backup source (Origem de backup). Na caixa Propagar local S3 do arquivo RDB, digite o caminho do Amazon S3 para o(s) arquivo(s). Se você tiver vários arquivos .rdb, digite o caminho para cada um em uma lista separada por vírgulas. O caminho do Amazon S3 parece-se com *myBucket/myFolder/myBackupFilename*.rdb.

- Como usar a AWS CLI

Se você usar a operação `create-cache-cluster` ou `create-replication-group`, use o parâmetro `--snapshot-arns` para especificar um ARN totalmente qualificado para cada arquivo .rdb. Por exemplo, `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. O ARN deve ser resolvido para os arquivos de backup que você armazenou no Amazon S3.

- Uso da API do ElastiCache

Se você usar a operação da API do ElastiCache `CreateCacheCluster` ou `CreateReplicationGroup`, use o parâmetro `SnapshotArns` para especificar um ARN totalmente qualificado para cada arquivo .rdb. Por exemplo, `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. O ARN deve ser resolvido para os arquivos de backup que você armazenou no Amazon S3.

Important

Ao criar um cluster do Redis (modo cluster habilitado), você precisa configurar cada grupo de nós (fragmento) no novo cluster ou grupo de replicação. Use o parâmetro `--node-group-configuration` (API: `NodeGroupConfiguration`) para fazer isso. Para ver mais informações, consulte:

- ILC: [create-replication-group](#) na AWS CLI Referência
- API: [CreateReplicationGroup](#) na Referência da API do ElastiCache

Durante o processo de criação do seu cluster, os dados no seu backup do Redis são gravados no cluster. Você pode monitorar o progresso visualizando as mensagens de eventos do ElastiCache. Para fazer isso, consulte o console do ElastiCache e escolha Cache Events (Eventos do cache). Você também pode usar a interface de linha de comando do ElastiCache da AWS ou a API do

ElastiCache para obter mensagens de eventos. Para obter mais informações, consulte [Visualização de eventos do ElastiCache](#).

Versões de mecanismos e atualização

Esta seção aborda as versões compatíveis do mecanismo Redis e como atualizar.

Tópicos

- [Versões de mecanismos e atualização](#)
- [Versões do ElastiCache para Redis com suporte](#)
- [Cronograma de fim de vida das versões do Redis](#)
- [Como atualizar as versões dos mecanismos](#)
- [Resolução de atualizações do mecanismo do Redis bloqueadas](#)
- [Principais diferenças de comportamento e compatibilidade da versão](#)

Versões de mecanismos e atualização

As versões do ElastiCache para Redis são identificadas com uma versão semântica que compreende um componente MAJOR e MINOR. Por exemplo, no Redis 6.2, a versão principal é a 6, a versão secundária é a 2. Ao operar clusters autoprojeto, o ElastiCache para Redis também expõe o componente PATCH, por exemplo, Redis 6.2.1, e a versão do patch é 1.

As versões PRINCIPAIS são para alterações incompatíveis da API e as versões SECUNDÁRIAS são para novas funcionalidades adicionadas de forma compatível com as versões anteriores. As versões do PATCH são para correções de bugs compatíveis com versões anteriores e mudanças não funcionais.

Gerenciamento de versões para ElastiCache Sem Servidor

O ElastiCache Sem Servidor aplica automaticamente a versão mais recente do software MINOR e PATCH ao cache, sem nenhum impacto ou tempo de inatividade sobre a aplicação. Não é necessária nenhuma ação de sua parte.

Quando uma nova versão MAJOR estiver disponível, o ElastiCache Sem Servidor vai enviar uma notificação no console e um evento no EventBridge. Você pode optar por atualizar o cache para a versão principal mais recente modificando o cache usando o console, a CLI ou a API e selecionando a versão mais recente do mecanismo.

Gerenciamento de versões para clusters do ElastiCache autoprojeto

Ao trabalhar com clusters do ElastiCache autoprojeto, você pode controlar quando o software que capacita o cluster de cache é atualizado para novas versões compatíveis pelo ElastiCache. É possível controlar quando atualizar o cache para as versões MAJOR, MINOR e PATCH mais recentes disponíveis. Você inicia atualizações de versão do mecanismo no seu cluster ou grupo de replicação, modificando-o e especificando uma nova versão do mecanismo.

Você pode controlar se e quando os softwares compatíveis com o protocolo que alimenta seu cluster de cache são atualizados para novas versões com suporte 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.

Você inicia atualizações de versão do mecanismo no seu cluster ou grupo de replicação, modificando-o e especificando uma nova versão do mecanismo. Para ver mais informações, consulte:

- [Modificação de clusters](#)
- [Modificação de um grupo de replicação](#)

Considerações sobre atualização ao trabalhar com clusters autoprotetidos

Note

As considerações a seguir só se aplicam ao atualizar clusters autoprotetidos. Elas não se aplicam ao ElastiCache Sem Servidor.

Ao atualizar um cluster autoprotetido, leve em consideração o seguinte

- O gerenciamento da versão do mecanismo foi desenvolvido para que você possa ter o máximo controle possível sobre a execução de patches. No entanto, o ElastiCache reserva-se o direito de executar patches no cluster em seu nome caso ocorra uma vulnerabilidade de segurança crítica no sistema ou software de cache, o que é improvável.
- A partir do Redis 6.0, o ElastiCache para Redis oferecerá uma única versão para cada versão secundária do Redis OSS, em vez de oferecer várias versões de patch.
- A partir do mecanismo Redis versão 5.0.6, você pode atualizar a versão do cluster com o mínimo de tempo de inatividade. O cluster estará disponível para leituras durante todo o processo de atualização e para gravações durante a maior parte da atualização, exceto durante a operação de failover que dura alguns segundos.
- Você também pode atualizar os clusters do ElastiCache com versões anteriores à 5.0.6. O processo envolvido é o mesmo, mas pode incorrer em tempo de failover mais longo durante a propagação do DNS (30 s a 1 m).
- A partir do Redis 7, o ElastiCache para Redis é compatível com a alternância entre o Redis (modo de cluster desabilitado) e o Redis (modo de cluster habilitado).
- O processo de atualização de mecanismo do Amazon ElastiCache for Redis foi projetado para fazer o melhor esforço no sentido de reter seus dados existentes e requer uma replicação do Redis para que seja bem-sucedido.

- Ao atualizar o mecanismo, o ElastiCache para Redis encerrará as conexões de clientes existentes. Para minimizar o tempo de inatividade durante as atualizações do mecanismo, recomendamos implementar [as práticas recomendadas para clientes do Redis](#) com novas tentativas de erro e recuo exponencial e as práticas recomendadas para [minimizar o tempo de inatividade durante a manutenção](#).
- Você não pode atualizar diretamente do Redis (modo cluster desabilitado) para o Redis (modo cluster habilitado) ao atualizar o mecanismo. O procedimento a seguir mostra como atualizar do Redis (modo cluster desabilitado) para o Redis (modo cluster habilitado).

Para atualizar de uma versão do mecanismo Redis (modo cluster desabilitado) para o Redis (modo cluster habilitado)

1. Faça um backup do seu cluster do Redis (modo cluster desabilitado) ou do grupo de replicação. Para obter mais informações, consulte [Realização de backups manuais](#).
 2. Use o backup para criar e propagar um cluster do Redis (modo cluster habilitado) com um fragmento (grupo de nós). Especifique a nova versão do mecanismo e habilite o modo de cluster ao criar o cluster ou o grupo de replicação. Para obter mais informações, consulte [Propagação de um novo cluster autoprojeto com um backup criado externamente](#).
 3. Exclua o antigo cluster do Redis (modo cluster desabilitado) ou o grupo de replicação. Para obter mais informações, consulte [Excluir um cluster](#) ou [Exclusão de um grupo de replicação](#).
 4. Escale o novo cluster ou grupo de replicação do Redis (modo cluster habilitado) para o número de fragmentos (grupos de nós) que você precisa. Para obter mais informações, consulte [Escalaabilidade de clusters no Redis \(modo cluster habilitado\)](#)
- Ao atualizar as principais versões do mecanismo, por exemplo, de 5.0.6 para 6.0, também é necessário selecionar um novo grupo de parâmetros que seja compatível com a nova versão do mecanismo.
 - Para clusters Redis de nó único e clusters Redis individuais com o Multi-AZ desabilitado, recomendamos que seja disponibilizada memória suficiente para o Redis, conforme descrito em [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#). Nesses casos, o primário não está disponível para solicitações de serviço durante o processo de atualização.
 - Para clusters Redis com o Multi-AZ habilitado, também recomendamos que você programe atualizações do mecanismo durante períodos de baixo tráfego de gravações recebidas. Ao atualizar para o Redis 5.0.6 ou posterior, o cluster primário continua disponível para solicitações de serviço durante o processo de atualização.

Os clusters e os grupos de replicação com vários fragmentos são processados e corrigidos da seguinte forma:

- Todos os estilhaços são processados em paralelo. Somente uma operação de atualização é realizada em um estilhaço por vez.
- Em cada fragmento, todas as réplicas são processadas antes do processamento da primária. Caso haja menos réplicas em um fragmento, a primária nesse fragmento pode ser processada antes da conclusão do processamento das réplicas em outros fragmentos.
- Em todos os fragmentos, os nós primários são processados em série. Somente um nó primário é atualizado por vez.
- Caso a criptografia esteja habilitada no cluster ou no grupo de replicação atual, não será possível atualizar para uma versão de mecanismo que não ofereça suporte à criptografia, como de 3.2.6 a 3.2.10.

Como atualizar as versões dos mecanismos

Você inicia atualizações de versão no seu cluster ou grupo de replicação modificando-o com o uso do console do ElastiCache, da AWS CLI ou da API do ElastiCache e especificando uma versão mais nova do mecanismo. Para obter mais informações, consulte os tópicos a seguir.

Como modificar clusters e grupos de replicação	
Clusters	Grupos de replicação
Usando o AWS Management Console	Usar a AWS Management Console
Usando o AWS CLI	Usar a AWS CLI
Usando a ElastiCache API	Uso da API do ElastiCache

Resolução de atualizações do mecanismo do Redis bloqueadas

Conforme mostrado na tabela a seguir, sua operação de atualização do mecanismo Redis será bloqueada se você tiver uma operação de expansão 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

Como resolver uma atualização bloqueada do mecanismo do Redis

- Execute um destes procedimentos:
 - Planeje sua operação de atualização do mecanismo Redis para a próxima janela de manutenção, desmarcando a caixa de seleção Apply immediately (Aplicar imediatamente).

Com a CLI, use `--no-apply-immediately`. Com a API, use `ApplyImmediately=false`.

- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de atualização do mecanismo Redis.
- Adicione a operação de expansão do Redis a essa modificação de cluster com a caixa de seleção Apply Immediately (Aplicar imediatamente) marcada.

Com a CLI, use `--apply-immediately`. Com a API, use `ApplyImmediately=true`.

Essa abordagem cancela efetivamente a atualização do mecanismo durante a próxima janela de manutenção, executando-a imediatamente.

Versões do ElastiCache para Redis com suporte

Os caches do ElastiCache Sem Servidor são compatíveis com as seguintes versões do Redis:

- [ElastiCache para Redis versão 7.0 \(aprimorado\)](#)

Os clusters do ElastiCache autoprojeto dão suporte às seguintes versões do Redis:

- [ElastiCache para Redis versão 7.0 \(aprimorado\)](#)
- [ElastiCache para Redis versão 7.0 \(aprimorado\)](#)
- [ElastiCache para Redis versão 6.2 \(aprimorado\)](#)
- [ElastiCache para Redis versão 6.0 \(aprimorado\)](#)
- [ElastiCache para Redis versão 5.0.6 \(aprimorado\)](#)
- [ElastiCache para Redis versão 5.0.5 \(obsoleto, use a versão 5.0.6\)](#)
- [ElastiCache para Redis versão 5.0.4 \(obsoleto, use a versão 5.0.6\)](#)
- [ElastiCache para Redis versão 5.0.3 \(obsoleto, use a versão 5.0.6\)](#)
- [ElastiCache para Redis versão 5.0.0 \(obsoleto, use a versão 5.0.6\)](#)
- [ElastiCache para Redis versão 4.0.10 \(aprimorado\)](#)
- [Versões que ultrapassaram o fim de vida útil \(EOL\) \(3.x\)](#)
- [Versões que ultrapassaram o fim de vida útil \(EOL\) \(2.x\)](#)

ElastiCache para Redis versão 7.0 (aprimorado)

Esta versão contém melhorias de desempenho que permitem que as workloads aumentem o throughput e diminuam as latências de operação. O 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 que os threads de E/S aprimorados agora não estão apenas lendo a entrada do cliente, mas também analisando a entrada no formato de comando binário do Redis. Em seguida, isso é encaminhado para o thread principal para execução, o que proporciona um ganho de desempenho. Padrão de acesso à memória do Redis 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 o ElastiCache em R7g.4xLarge baseado em Graviton3 ou maior, os clientes podem

alcançar mais de um milhão de solicitações por segundo por nó. Com as melhorias de desempenho feitas no ElastiCache para Redis v7.1, os clientes podem obter até 100% mais throughput e latência P99 50% menor em relação ao ElastiCache para Redis v7.0. Esses aprimoramentos são habilitados em tamanhos de nós com pelo menos oito núcleos físicos (2x1large no Graviton e 4x1large no x86), independentemente do tipo de CPU, e não exigem alterações no cliente.

Note

O ElastiCache v7.1 é compatível com o OSS Redis v7.0.

ElastiCache para Redis versão 7.0 (aprimorado)

O ElastiCache para Redis 7.0 adiciona várias melhorias e suporte para novas funcionalidades:

- [Funções do Redis](#): o ElastiCache para Redis 7 adiciona suporte às funções do Redis e fornece uma experiência gerenciada que permite que os desenvolvedores executem [scripts LUA](#) com a lógica da aplicação armazenada no cluster do ElastiCache, sem exigir que os clientes reenviem os scripts para o servidor com cada conexão.
- [Melhorias na ACL](#): o ElastiCache para Redis 7 adiciona suporte para a próxima versão das listas de controle de acesso (ACLs) do Redis. Com o ElastiCache para Redis 7, os clientes agora podem especificar vários conjuntos de permissões em chaves ou espaços de chave específicos no Redis.
- [Pub/Sub fragmentado](#): o ElastiCache para Redis 7 adiciona suporte para executar a funcionalidade Redis Pub/Sub de forma fragmentada ao executar o ElastiCache no modo de cluster habilitado (CME). Os recursos do Redis Pub/Sub permitem que os editores enviem mensagens para qualquer número de assinantes em um canal. Com o Amazon ElastiCache para Redis 7, os canais são vinculados a um fragmento no cluster do ElastiCache, eliminando a necessidade de propagar as informações do canal entre os fragmentos, resultando em maior escalabilidade.
- Multiplexação de E/S aprimorada: o ElastiCache para Redis versão 7 apresenta a multiplexação de E/S aprimorada, que proporciona maior throughput e menor latência para workloads de alto throughput com muitos clientes conectados simultaneamente a um cluster do ElastiCache. Por exemplo, ao usar um cluster de nós r6g.xlarge e executar 5,2 mil clientes simultâneos, você pode obter até 72% de aumento no throughput (operações de leitura e gravação por segundo) e redução de até 71% na latência P99, em comparação com o ElastiCache para Redis versão 6.

Para obter mais informações sobre o Redis 7.0, consulte as [Notas de release do Redis 7.0](#) em Redis no GitHub.

ElastiCache para Redis versão 6.2 (aprimorado)

O ElastiCache para Redis 6.2 inclui aprimoramentos de performance para clusters habilitados para TLS que usam nós do tipo x86 com 8 vCPUs ou mais, ou nós dos tipos Graviton2 com 4 vCPUs ou mais. Esses aprimoramentos melhoram o throughput e reduzem o tempo de estabelecimento da conexão do cliente ao descarregar a criptografia para outras vCPUs. Com o Redis 6.2, você também pode gerenciar o acesso aos canais Pub/Sub com as regras da Access Control List (ACL – Lista de controle de acesso).

Com essa versão, também apresentamos o suporte para classificação de dados em níveis em nós de cluster contendo SSDs NVMe conectados localmente. Para obter mais informações, consulte [Classificação de dados em níveis](#).

A versão 6.2.6 do mecanismo Redis também é compatível com o JavaScript Object Notation (JSON), que é uma forma simples e sem esquema de codificar conjuntos de dados complexos em clusters do Redis. Com o suporte ao JSON, é possível aproveitar a performance e as APIs Redis para aplicações que operam em JSON. Para obter mais informações, consulte [Conceitos básicos do JSON](#). As métricas relacionadas ao JSON `JsonBasedCmds` e `JsonBasedCmdsLatency`, que são incorporadas ao CloudWatch para monitorar o uso desses tipos de dados. Para obter mais informações, consulte [Métricas para Redis](#).

Você especifica a versão do mecanismo usando 6.2. O ElastiCache para Redis invocará automaticamente a versão de patch preferencial do Redis 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 iniciado com a versão de patch preferencial atual disponível do Redis 6.2 no momento da criação/modificação. Especificar a versão 6.x do mecanismo na API resultará na versão secundária mais recente do Redis 6.

Para clusters 6.0 existentes, você pode optar pela próxima atualização automática da versão secundária definindo o parâmetro `AutoMinorVersionUpgrade` como `yes` nas APIs `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` ou `ModifyReplicationGroup`. O ElastiCache para Redis atualizará a versão secundária dos clusters 6.0 existentes para 6.2 usando atualizações com autoatendimento. Para mais informações, consulte [Atualizações com autoatendimento no Amazon ElastiCache](#).

Ao chamar a API `DescribeCacheEngineVersions`, o valor do parâmetro `EngineVersion` será definido como 6.2 e a versão efetiva do mecanismo com a versão do patch será retornada no campo `CacheEngineVersionDescription`.

Para mais informações sobre a versão do Redis 6.2, consulte as [Notas de release do Redis 6.2](#) em Redis no GitHub.

ElastiCache para Redis versão 6.0 (aprimorado)

O Amazon ElastiCache para Redis apresenta a próxima versão do mecanismo Redis, que inclui [autenticação de usuários com controle de acesso baseado em função](#), cache do lado do cliente e melhorias operacionais significativas.

A partir do Redis 6.0, o ElastiCache para Redis oferecerá uma única versão para cada versão secundária do Redis OSS, em vez de oferecer várias versões de patch. O ElastiCache para Redis gerenciará automaticamente a versão de patch de seus clusters de cache em execução, garantindo melhor performance e segurança aprimorada.

Você também pode optar pela atualização automática da próxima versão secundária definindo o parâmetro `AutoMinorVersionUpgrade` para `yes`. O ElastiCache para Redis gerenciará a atualização de versão secundária por meio de atualizações com autoatendimento. Para obter mais informações, consulte [Atualizações de serviço em ElastiCache](#).

Você especifica a versão do mecanismo usando `6.0`. O ElastiCache para Redis invocará automaticamente a versão de patch preferencial do Redis 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 iniciado com a versão de patch preferencial atual disponível do Redis 6.0 no momento da criação/modificação. Qualquer solicitação com um valor específico de versão de patch será rejeitada, uma exceção será lançada e o processo falhará.

Ao chamar a API `DescribeCacheEngineVersions`, o valor do parâmetro `EngineVersion` será definido como `6.0` e a versão efetiva do mecanismo com a versão do patch será retornada no campo `CacheEngineVersionDescription`.

Para obter mais informações sobre o Redis 6.0, consulte as [Notas de release do Redis 6.0](#) em Redis no GitHub.

ElastiCache para Redis versão 5.0.6 (aprimorado)

O Amazon ElastiCache para Redis apresenta a próxima versão do mecanismo do 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 denominado `rename-commands` que permite renomear comandos do Redis possivelmente perigosos e caros que podem causar a perda acidental de dados, como `FLUSHALL` ou `FLUSHDB`. Isso é semelhante à configuração `rename-command` no Redis de software livre. No entanto, o ElastiCache melhorou a experiência ao fornecer um fluxo totalmente gerenciado. As alterações de nomes de comando são aplicadas imediatamente e propagadas automaticamente em todos os nós no cluster que contêm a lista de comandos. Não há nenhuma intervenção necessária de sua parte, como a reinicialização de nós.

Os exemplos a seguir demonstram como modificar os grupos de parâmetros existentes. Eles incluem o parâmetro `rename-commands`, que é uma lista separada por espaços de comandos que você deseja renomear:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

Neste exemplo, o parâmetro `rename-commands` é usado para renomear o comando `flushall` como `restrictedflushall`.

Para renomear vários comandos, use o seguinte:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall flushdb restrictedflushdb'" --region region
```

Para reverter qualquer alteração, execute novamente o comando e exclua qualquer valor renomeado na lista `ParameterValue` que você deseja manter, conforme mostrado a seguir:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

Nesse caso, o comando `flushall` é renomeado para `restrictedflushall` e qualquer outro comando renomeado revertido para seus nomes de comando originais.

Note

Ao renomear comandos, você está restrito às seguintes limitações:

- Todos os comandos renomeados devem ser alfanuméricos.
- O tamanho máximo de novos nomes de comando é de 20 caracteres alfanuméricos.
- Ao renomear comandos, certifique-se de que você atualize o grupo de parâmetros associado ao seu cluster.
- Para evitar totalmente o uso de um comando, use a palavra-chave `blocked`, conforme mostrado a seguir:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall blocked'" --region region
```

Para obter mais informações sobre as alterações de parâmetro e uma lista de comandos que são qualificados para serem renomeados, consulte [Alterações de parâmetros do Redis 5.0.3](#).

- Redis Streams: essa opção modela uma estrutura de dados de log que permite que os produtores anexem novos itens em tempo real. Também permite que os consumidores leiam mensagens com ou sem bloqueio. Os fluxos também permitem grupos de consumidores, que representam um grupo de clientes para cooperativamente consumir diferentes partes do mesmo fluxo de mensagens, semelhante ao [Apache Kafka](#). Para obter mais informações, consulte [Introdução aos fluxos Redis](#).
- Oferece suporte para uma família de comandos de fluxo, como os `XADD`, `XRANGE` e `XREAD`. Para obter mais informações, consulte [Comandos dos fluxos Redis](#).
- Diversos parâmetros novos e renomeados. Para obter mais informações, consulte [Alterações de parâmetros do Redis 5.0.0](#).
- Uma nova métrica do Redis, `StreamBasedCmds`.
- Tempo de snapshot um pouco mais rápido para nós Redis.

⚠ Important

O Amazon ElastiCache para Redis portou de volta duas correções críticas de bugs da [versão 5.0.1 de código aberto do Redis](#). Eles estão listados a seguir:

- RESTAURAR resposta incompatível quando determinadas chaves já expiraram.
- O comando XCLAIM pode retornar uma entrada errada ou dessincronizar o protocolo.

Ambas as correções de erro estão incluídas no suporte do ElastiCache para Redis para a versão 5.0.0 do mecanismo do Redis e serão consumidas em atualizações de versões futuras.

Para obter mais informações, consulte as [Notas de release do Redis 5.0.6](#) em Redis no GitHub.

ElastiCache para Redis versão 5.0.5 (obsoleto, use a versão 5.0.6)

O Amazon ElastiCache para Redis apresenta a próxima versão do mecanismo do Redis. Ele inclui alterações de configuração online do ElastiCache para Redis de clusters de failover automático durante todas as operações planejadas. Agora é possível dimensionar seu cluster, atualizar a versão do mecanismo Redis e aplicar patches e atualizações de manutenção enquanto o cluster permanece online e continua atendendo às solicitações recebidas. Ela também inclui correções de erros.

Para obter mais informações, consulte as [Notas de release do Redis 5.0.5](#) em Redis no GitHub.

ElastiCache para Redis versão 5.0.4 (obsoleto, use a versão 5.0.6)

O Amazon ElastiCache para Redis apresenta a próxima versão do mecanismo do Redis com suporte do 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 release do Redis 5.0.4](#) em Redis no GitHub.

ElastiCache para Redis versão 5.0.3 (obsoleto, use a versão 5.0.6)

O Amazon ElastiCache para Redis apresenta a próxima versão do mecanismo do Redis compatível com o Amazon ElastiCache, que inclui correções de bugs.

ElastiCache para Redis versão 5.0.0 (obsoleto, use a versão 5.0.6)

O Amazon ElastiCache para Redis apresenta a próxima versão principal do mecanismo do Redis com suporte pelo Amazon ElastiCache. O ElastiCache para Redis 5.0.0 oferece suporte para as seguintes melhorias:

- Redis Streams: essa opção modela uma estrutura de dados de log que permite que os produtores anexem novos itens em tempo real. Também permite que os consumidores leiam mensagens com ou sem bloqueio. Os fluxos também permitem grupos de consumidores, que representam um grupo de clientes para cooperativamente consumir diferentes partes do mesmo fluxo de mensagens, semelhante ao [Apache Kafka](#). Para obter mais informações, consulte [Introdução aos fluxos Redis](#).
- Oferece suporte para uma família de comandos de fluxo, como os XADD, XRANGE e XREAD. Para obter mais informações, consulte [Comandos dos fluxos Redis](#).
- Diversos parâmetros novos e renomeados. Para obter mais informações, consulte [Alterações de parâmetros do Redis 5.0.0](#).
- Uma nova métrica do Redis, `StreamBasedCmds`.
- Tempo de snapshot um pouco mais rápido para nós Redis.

ElastiCache para Redis versão 4.0.10 (aprimorado)

O Amazon ElastiCache para Redis apresenta a próxima versão principal do mecanismo do Redis com suporte pelo Amazon ElastiCache. O ElastiCache para Redis 4.0.10 oferece suporte para as seguintes melhorias:

- Redimensionamento e criptografia de cluster online em uma única versão do ElastiCache para Redis. Para ver mais informações, consulte:
 - [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#)
 - [Refragmentação e rebalanceamento de fragmentos online para o Redis \(modo cluster habilitado\)](#)

- [Segurança de dados no Amazon ElastiCache](#)
- Diversos novos parâmetros. Para obter mais informações, consulte [Alterações de parâmetros do Redis 4.0.10](#).
- Suporte para a família de comandos de memória, como MEMORY. Para obter mais informações, consulte [Comandos do Redis](#) (pesquisar em MEMÓRIA).
- Suporte para desfragmentação de memória online, permitindo assim a utilização mais eficiente da memória e uma quantidade maior de memória disponível para seus dados.
- Suporte a liberações e exclusões assíncronas. O ElastiCache para Redis oferece suporte a comandos como UNLINK, FLUSHDB e FLUSHALL para execução em um thread diferente do principal. Fazer isso ajuda a melhorar o desempenho e os tempos de resposta dos aplicativos, liberando a memória de maneira assíncrona.
- Uma nova métrica do Redis, ActiveDefragHits. Para obter mais informações, consulte [Métricas do Redis](#).

Os usuários do Redis (modo cluster desabilitado) que executam a versão 3.2.10 do Redis podem usar o console para atualizar seus clusters por meio da atualização online.

Comparando o redimensionamento de cluster do ElastiCache para Redis e o suporte à criptografia

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.

** Obrigatório para aplicativos compatíveis com FedRAMP, HIPAA e PCI DSS. Para obter mais informações, consulte [Validação de conformidade para o Amazon ElastiCache](#).

Versões que ultrapassaram o fim de vida útil (EOL) (3.x)

ElastiCache para Redis versão 3.2.10 (aprimorado)

O Amazon ElastiCache para Redis apresenta a próxima versão principal do mecanismo do Redis com suporte pelo Amazon ElastiCache. O ElastiCache para Redis 3.2.10 apresenta o redimensionamento online de cluster para adicionar ou remover fragmentos do cluster enquanto continua a atender as solicitações recebidas de E/S. Os usuários do ElastiCache para Redis 3.2.10 têm todas as funcionalidades das versões anteriores do Redis, exceto a capacidade de criptografar seus dados. Esse recurso está atualmente disponível apenas na versão 3.2.6.

Comparação das versões 3.2.6 e 3.2.10 do ElastiCache para Redis

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.

** Obrigatório para aplicativos compatíveis com FedRAMP, HIPAA e PCI DSS. Para obter mais informações, consulte [Validação de conformidade para o Amazon ElastiCache](#).

Para ver mais informações, consulte:

- [Refragmentação e rebalanceamento de fragmentos online para o Redis \(modo cluster habilitado\)](#)
- [Redimensionamento de cluster on-line](#)

ElastiCache para Redis versão 3.2.6 (aprimorado)

O Amazon ElastiCache para Redis apresenta a próxima versão principal do mecanismo do Redis com suporte pelo Amazon ElastiCache. Os usuários do ElastiCache para Redis 3.2.6 têm todas as funcionalidades das versões anteriores do Redis, além da opção de criptografar seus dados. Para ver mais informações, consulte:

- [Criptografia em trânsito \(TLS\) do ElastiCache](#)

- [Criptografia em repouso no ElastiCache](#)
- [Validação de conformidade para o Amazon ElastiCache](#)

ElastiCache para Redis versão 3.2.4 (aprimorado)

O Amazon ElastiCache para Redis versão 3.2.4 apresenta a próxima versão principal do mecanismo do Redis com suporte pelo Amazon ElastiCache. Os usuários do ElastiCache para Redis 3.2.4 têm toda a funcionalidade das versões Redis anteriores disponível, além da opção para executar no modo cluster ou no modo sem cluster. A tabela a seguir fornece um resumo.

Comparação do Redis 3.2.4 modo cluster e modo sem cluster

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 3.2.4 oferece suporte para a indexação geoespacial por meio de seis comandos GEO. Para obter mais informações, consulte a documentação dos comandos GEO* do Redis [Comandos do Redis: GEO](#), na página Redis Commands (filtrada por GEO).

Para obter informações sobre recursos adicionais do Redis 3, consulte as [notas de versão do Redis 3.2](#) e as [notas de release do Redis 3.0](#).

Atualmente, o Redis (modo cluster habilitado) gerenciado pelo ElastiCache não oferece suporte aos seguintes recursos do Redis 3.2:

- Migração de réplica
- Rebalanceamento do cluster
- Depurador Lua

O ElastiCache desabilita os seguintes comandos de gerenciamento do Redis 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 3.2.4, consulte [Alterações de parâmetros do Redis 3.2.4](#).

Versões que ultrapassaram o fim de vida útil (EOL) (2.x)

ElastiCache para Redis versão 2.8.24 (aprimorado)

Melhorias do Redis adicionadas desde a versão 2.8.23 incluem correções de bugs e o registro em log de endereços incorretos de acesso à memória. Para obter mais informações, consulte as [Notas de release do Redis 2.8](#).

ElastiCache para Redis versão 2.8.23 (aprimorado)

Melhorias do Redis adicionadas desde a versão 2.8.22 incluem correções de bugs. Para obter mais informações, consulte as [Notas de release do Redis 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 2.8.23, consulte [Parâmetros adicionados do Redis 2.8.23 \(aprimorado\)](#) no Guia do usuário do ElastiCache.

ElastiCache para Redis versão 2.8.22 (aprimorado)

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.
- Suporte a duas novas métricas do CloudWatch.
 - `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 [Monitoramento do uso com métricas do CloudWatch](#).

- Uma série de correções críticas de bugs no comportamento PSYNC de replicação. Para obter mais informações, consulte as [Notas de release do Redis 2.8](#).
- Para manter a performance de replicação aprimorado em grupos de replicação multi-AZ e para maior estabilidade de cluster, não há mais suporte para réplicas não ElastiCache.
- Para melhorar a consistência de dados entre o cluster primário e as réplicas em um grupo de replicação, as réplicas não removem mais as chaves independentemente do cluster primário.
- As variáveis de configuração do Redis `appendonly` e `appendfsync` não são compatíveis no Redis versão 2.8.22 e posterior.
- Em situações de pouca memória, os clientes com um grande buffer de saída podem ser desconectados de um cluster de réplicas. Se desconectado, o cliente deverá se reconectar. Essas situações são mais prováveis para clientes PUBSUB.

ElastiCache para Redis versão 2.8.21

Melhorias do Redis adicionadas desde a versão 2.8.19 incluem várias correções de bugs. Para obter mais informações, consulte as [Notas de release do Redis 2.8](#).

ElastiCache para Redis versão 2.8.19

Melhorias do Redis adicionadas desde a versão 2.8.6 incluem o seguinte:

- Suporte para HyperLogLog. Para obter mais informações, consulte [Nova 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 nós de réplica, a SYNC mestre falhará se um processo filho de salvamento (`bgsave`) for anulado.
- Suporte para a métrica do CloudWatch `HyperLogLogBasedCommands`. Para obter mais informações, consulte [Métricas para Redis](#).

ElastiCache para Redis versão 2.8.6

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 para Redis versão 2.6.13

O Redis versão 2.6.13 foi a versão inicial do Redis com suporte pelo Amazon ElastiCache para Redis. O Multi-AZ não tem suporte no Redis 2.6.13.

Cronograma de fim de vida das versões do Redis

Esta seção define as datas de fim de vida útil (EOL) para as versões principais mais antigas à medida que são anunciadas. Ela ajudará você a tomar decisões sobre versões e atualizações no futuro.

Note

As versões de patch do ElastiCache para Redis 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 data de EOL anunciada, bem como a versão de destino de atualização recomendada.

EOL passado

Versões secundárias de origem	Destino de atualização recomendado	Data de EOL
3.2.4, 3.2.6 e 3.2.10	Versão 6.2 ou superior	31 de julho de 2023

Versões secundárias de origem	Destino de atualização recomendado	Data de EOL
	<p> Note</p> <p>Para as regiões US-ISO-EAST-1, US-ISO-WEST-1 e US-ISOB-EAST-1, recomendamos a versão 5.0.6 ou superior.</p>	
V 2.8.24, 2.8.23, 2.8.22, 2.8.21, 2 2.8.19, 2.8.12, 2.8.6, 2.6.13	<p>Versão 6.2 ou superior</p> <p> Note</p> <p>Para as regiões US-ISO-EAST-1, US-ISO-WEST-1 e US-ISOB-EAST-1, recomendamos a versão 5.0.6 ou superior.</p>	13 de janeiro de 2023

Como atualizar as versões dos mecanismos

Você inicia atualizações de versão no seu cluster ou grupo de replicação modificando-o com o uso do console do ElastiCache, da AWS CLI ou da API do ElastiCache e especificando uma versão mais nova do mecanismo. Para obter mais informações, consulte os tópicos a seguir.

Como modificar clusters e grupos de replicação	
Clusters	Grupos de replicação
Usando o AWS Management Console	Usar a AWS Management Console
Usando o AWS CLI	Usar a AWS CLI
Usando a ElastiCache API	Uso da API do ElastiCache

Resolução de atualizações do mecanismo do Redis bloqueadas

Conforme mostrado na tabela a seguir, sua operação de atualização do mecanismo Redis será bloqueada se você tiver uma operação de expansão 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

Como resolver uma atualização bloqueada do mecanismo do Redis

- Execute um destes procedimentos:
 - Planeje sua operação de atualização do mecanismo Redis para a próxima janela de manutenção, desmarcando a caixa de seleção `Apply immediately` (Aplicar imediatamente).

Com a CLI, use `--no-apply-immediately`. Com a API, use `ApplyImmediately=false`.

- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de atualização do mecanismo Redis.
- Adicione a operação de expansão do Redis a essa modificação de cluster com a caixa de seleção Apply Immediately (Aplicar imediatamente) marcada.

Com a CLI, use `--apply-immediately`. Com a API, use `ApplyImmediately=true`.

Essa abordagem cancela efetivamente a atualização do mecanismo durante a próxima janela de manutenção, executando-a imediatamente.

Principais diferenças de comportamento e compatibilidade da versão

Important

A página a seguir é estruturada para indicar todas as diferenças de incompatibilidade entre versões e informar você sobre quaisquer considerações que deva fazer ao atualizar para versões mais recentes. Essa lista inclui todos os problemas de incompatibilidade de versão que você possa encontrar ao atualizar.

Você pode fazer o upgrade diretamente da sua versão atual do Redis para a versão mais recente disponível, sem a necessidade de atualizações sequenciais. Por exemplo, você pode atualizar diretamente da versão 3.0 do Redis para a versão 7.0.

As versões do Redis são identificadas com uma versão semântica que compreende um componente MAJOR, MINOR e PATCH. Por exemplo, no Redis 4.0.10, a versão principal é a 4, a versão secundária é a 0 e a versão do patch é a 10. Esses valores geralmente são incrementados com base nas seguintes convenções:

- As versões MAJOR (PRINCIPAL) são para alterações incompatíveis com a API
- As versões MINOR (SECUNDÁRIA) são para novas funcionalidades adicionadas de forma compatível com as versões anteriores
- As versões do PATCH são para correções de bugs compatíveis com versões anteriores e mudanças não funcionais

Recomendamos sempre permanecer na versão mais recente do patch dentro de uma determinada versão MAJOR.MINOR para obter as melhorias mais recentes de performance e estabilidade.

A partir do Redis 6.0, o ElastiCache para Redis oferecerá uma única versão para cada versão secundária do Redis OSS, em vez de oferecer várias versões de patch. O ElastiCache para Redis gerenciará automaticamente a versão de patch de seus clusters de cache em execução, garantindo melhor performance 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 o ElastiCache expande a disponibilidade para uma nova região da AWS, o ElastiCache para Redis oferece suporte às duas versões MAJOR.MINOR mais recentes à época para a nova região. Por exemplo, se uma nova região da AWS for iniciada e as versões MAJOR.MINOR mais recentes do ElastiCache para Redis forem 7.0 e 6.2, o ElastiCache para Redis oferecerá suporte às versões 7.0 e 6.2 na nova região da AWS. À medida que novas versões MAJOR.MINOR do ElastiCache para Redis forem lançadas, o ElastiCache adicionará suporte às versões recém-lançadas do ElastiCache para Redis. Para saber mais sobre como escolher regiões para o ElastiCache, consulte [Escolher regiões e zonas de disponibilidade](#).

Ao fazer uma atualização que abrange versões principais ou secundárias, considere a lista a seguir, que inclui comportamento e alterações incompatíveis com versões anteriores lançadas com o Redis ao longo do tempo.

Alterações de comportamento do Redis 7.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis 7.0](#).

- `SCRIPT LOAD` e `SCRIPT FLUSH` não são mais propagados para réplicas. Se você precisar ter alguma durabilidade para scripts, recomendamos que você considere usar as [funções do Redis](#).
- Os canais Pubsub agora estão bloqueados por padrão para novos usuários de ACL.
- O comando `STRALGO` foi substituído pelo comando `LCS`.
- O formato de `ACL GETUSER` foi alterado para que todos os campos mostrem o padrão de cadeia de caracteres de acesso padrão. Se você teve automação usando `ACL GETUSER`, verifique se ela funcionará com qualquer um dos formatos.
- As categorias de ACL para `SELECT`, `WAIT`, `ROLE`, `LASTSAVE`, `READONLY`, `READWRITE` e `ASKING` foram alteradas.
- O comando `INFO` agora mostra estatísticas de comando por subcomando, em vez de nos comandos do contêiner de nível superior.

- Os valores de retorno dos comandos LPOP, RPOP, ZPOPMIN e ZPOPMAX foram alterados em determinados casos de borda. Se você usar esses comandos, verifique as notas de release e avalie se foi afetado.
- Os comandos SORT e SORT_R0 agora exigem acesso a todo o espaço de teclas para usar os argumentos GET e BY.

Alterações de comportamento do Redis 6.2 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis 6.2](#).

- Os sinalizadores ACL dos comandos TIME, ECHO, ROLE e LASTSAVE foram alterados. Isso pode fazer com que comandos que antes tinham permissão sejam rejeitados e vice-versa.

Note

Nenhum desses comandos modifica ou concede acesso aos dados.

- Ao atualizar do Redis 6.0, a ordem dos pares chave/valor retornados de uma resposta de mapa para um script lua é alterada. Se os seus scripts usarem `redis.setresp()` ou retornarem um mapa (novo no Redis 6.0), considere as implicações que o script pode quebrar nas atualizações.

Alterações de comportamento do Redis 6.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis 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 parâmetro `AutoMinorVersionUpgrade` como `sim` e o ElastiCache para Redis 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 mais informações, consulte [Atualizações com autoatendimento no ElastiCache](#).

Alterações de comportamento do Redis 5.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis 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 reverter para o comportamento anterior que só está disponível no ElastiCache para Redis 5.0.
- Se você estiver atualizando a partir do Redis 4.0, alguns comandos em scripts LUA retornarão argumentos em uma ordem diferente daquela em versões anteriores. No Redis 4.0, o Redis ordenaria algumas respostas lexograficamente para tornar as respostas determinísticas, essa ordem não é aplicada quando os scripts são replicados por efeitos.
- No Redis 5.0.3 e posterior, o ElastiCache para Redis descarregará alguns trabalhos de E/S para núcleos em segundo plano em tipos de instância com mais de quatro VCPUs. Isso pode alterar as características de performance do Redis e alterar os valores de algumas métricas. Para obter mais informações, consulte [Quais métricas devo monitorar?](#) para entender se você precisa alterar quais métricas você assiste.

Alterações de comportamento do Redis 4.0 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte as [Notas de versão do Redis 4.0](#).

- O log lento agora registra dois argumentos adicionais, o nome e o endereço do cliente. Essa alteração deve ser compatível com versões anteriores, a menos que você confie explicitamente em cada entrada de log lenta contendo 3 valores.
- O CLUSTER NODES agora retorna um formato ligeiramente diferente, que não é compatível com versões anteriores. Recomendamos que os clientes não usem esse comando para aprender sobre os nós presentes em um cluster e, em vez disso, eles usem CLUSTER SLOTS.

EOL passado

Alterações de comportamento do Redis 3.2 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte [Notas de versão do Redis 3.2](#).

- Não há mudanças de compatibilidade a serem destacadas para esta versão.

Para obter mais informações, consulte [Cronograma de fim de vida das versões do Redis](#).

Alterações de comportamento do Redis 2.8 e alterações incompatíveis com versões anteriores

Para obter uma lista completa de alterações, consulte [Notas de versão do Redis 2.8](#).

- Começando com o Redis 2.8.22, o Redis AOF não é mais compatível com o ElastiCache para Redis. Recomendamos o uso do MemoryDB quando os dados precisam ser persistidos de forma duradoura.
- A partir do Redis 2.8.22, o ElastiCache para Redis não oferece mais suporte à anexação de réplicas a primárias hospedadas no 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 6.0 como uma alternativa às réplicas externas.
- Os comandos TTL e PTTL agora retornarão -2 se a chave não existir e -1 se existir, mas não tiver expiração associada. O Redis 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 versões do Redis](#).

Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache

Abaixo, é possível encontrar práticas recomendadas para o Amazon ElastiCache. Seguir essas práticas melhora o desempenho e aumenta a confiabilidade do cache.

Tópicos

- [Trabalhar com o Redis](#)
- [Práticas recomendadas com clientes do Redis](#)
- [Práticas recomendadas durante o trabalho com clusters autoprojeto](#)
- [Práticas recomendadas do Redis](#)
- [Estratégias de armazenamento em cache](#)

Trabalhar com o Redis

Abaixo, é possível encontrar informações sobre a interface do Redis no ElastiCache.

Tópicos

- [Comandos do Redis compatíveis e restritos](#)
- [Configuração e limites do Redis](#)

Comandos do Redis compatíveis e restritos

Comandos do Redis compatíveis

Comandos do Redis compatíveis

Os comandos do Redis a seguir são compatíveis com caches sem servidor. Além desses comandos, estes [Comandos compatíveis com o JSON do Redis](#) 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 leitura para uma conexão com um nó de réplica do Redis Cluster.

[Saiba mais](#)

- READWRITE

Permite consultas de leitura e gravação para uma conexão com um nó de réplica do Redis 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

Cumprimenta o servidor do Redis.

[Saiba mais](#)

- PING

Retorna a resposta de atividade do servidor.

[Saiba mais](#)

- QUIT

Fecha a conexão.

[Saiba mais](#)

- RESET

Redefine a conexão.

[Saiba mais](#)

- SELECT

Altera o banco de dados selecionado.

[Saiba mais](#)

Comandos genéricos

- COPY

Copia o valor de uma chave para uma nova chave.

[Saiba mais](#)

- DEL

Exclui uma ou mais chaves.

[Saiba mais](#)

- DUMP

Retorna uma representação serializada do valor armazenado em uma chave.

[Saiba mais](#)

- EXISTS

Determina se existe uma ou mais chaves.

[Saiba mais](#)

- EXPIRE

Define o tempo de expiração de uma chave em segundos.

[Saiba mais](#)

- EXPIREAT

Define o tempo de expiração de uma chave como um carimbo de data e hora Unix.

[Saiba mais](#)

- EXPIRETIME

Retorna o tempo de expiração de uma chave como um carimbo de data e hora Unix.

[Saiba mais](#)

- PERSIST

Remove o tempo de expiração de uma chave.

[Saiba mais](#)

- PEXPIRE

Define o tempo de expiração de uma chave em milissegundos.

[Saiba mais](#)

- PEXPIREAT

Define o tempo de expiração de uma chave como um carimbo de data e hora Unix em milissegundos.

[Saiba mais](#)

- PEXPIRETIME

Retorna o tempo de expiração de uma chave como um carimbo de data e hora Unix em milissegundos.

[Saiba mais](#)

- PTTL

Define o tempo de expiração de uma chave em milissegundos.

[Saiba mais](#)

- RANDOMKEY

Retorna um nome de chave aleatório do banco de dados.

[Saiba mais](#)

- RENAME

Renomeia uma chave e substitui o destino.

[Saiba mais](#)

- RENAMENX

Só renomeia uma chave quando o nome da chave de destino não existe.

[Saiba mais](#)

- RESTORE

Cria uma chave a partir da representação serializada de um valor.

[Saiba mais](#)

- SCAN

Itera os nomes de chaves no banco de dados.

[Saiba mais](#)

- SORT

Classifica os elementos em uma lista, conjunto ou conjunto classificado, armazenando o resultado como opção.

[Saiba mais](#)

- SORT_RO

Retorna os elementos classificados de uma lista, um conjunto ou um conjunto classificado.

[Saiba mais](#)

- TOUCH

Retornará o número de chaves existentes dentre as especificadas depois de atualizar a hora em que foram acessadas pela última vez.

[Saiba mais](#)

- TTL

Retorna o tempo de expiração de uma chave em segundos.

[Saiba mais](#)

- TYPE

Determina o tipo de valor armazenado em uma chave.

[Saiba mais](#)

- UNLINK

Exclui de maneira assíncrona uma ou mais chaves.

[Saiba mais](#)

Comandos geoespaciais

- GEOADD

Adiciona um ou mais membros a um índice geoespacial. A chave será criada se não existir.

[Saiba mais](#)

- GEODIST

Retorna a distância entre dois membros de um índice geoespacial.

[Saiba mais](#)

- GEOHASH

Retorna membros de um índice geoespacial como strings geohash.

[Saiba mais](#)

- GEOPOS

Retorna a longitude e a latitude de membros de um índice geoespacial.

[Saiba mais](#)

- GEORADIUS

Consulta um índice geoespacial para membros a uma distância de uma coordenada e, como opção, armazena o resultado.

[Saiba mais](#)

- GEORADIUS_RO

Retorna membros de um índice geoespacial que estejam a uma distância de uma coordenada.

[Saiba mais](#)

- GEORADIUSBYMEMBER

Consulta um índice geoespacial para membros a uma distância de um membro e, como opção, armazena o resultado.

[Saiba mais](#)

- GEORADIUSBYMEMBER_RO

Retorna membros de um índice geoespacial que estejam a uma distância de um membro.

[Saiba mais](#)

- GEOSEARCH

Consulta um índice geoespacial para membros dentro de uma área de uma caixa ou círculo.

[Saiba mais](#)

- GEOSEARCHSTORE

Consulta um índice geoespacial para membros dentro de uma área de uma caixa ou círculo e, como opção, armazena o resultado.

[Saiba mais](#)

Comandos de hash

- HDEL

Exclui um ou mais campos e os valores de um hash. Exclui o hash se nenhum campo permanecer.

[Saiba mais](#)

- HEXISTS

Determina se existe um campo em um hash.

[Saiba mais](#)

- HGET

Retorna o valor de um campo em um hash.

[Saiba mais](#)

- HGETALL

Retorna todos os campos e valores em um hash.

[Saiba mais](#)

- HINCRBY

Incrementa o valor inteiro de um campo em um hash por um número. Usará 0 como valor inicial se o campo não existir.

[Saiba mais](#)

- HINCRBYFLOAT

Incrementa o valor de ponto flutuante de um campo por um número. Usará 0 como valor inicial se o campo não existir.

[Saiba mais](#)

- HKEYS

Retorna todos os campos em um hash.

[Saiba mais](#)

- HLEN

Retorna o número de chaves em um hash.

[Saiba mais](#)

- HMGET

Retorna os valores de todos os campos em um hash.

[Saiba mais](#)

- HMSET

Define os valores de vários campos.

[Saiba mais](#)

- HRANDFIELD

Retorna um ou mais campos aleatórios de um hash.

[Saiba mais](#)

- HSCAN

Itera campos e valores de um hash.

[Saiba mais](#)

- HSET

Cria ou modifica o valor de um campo em um hash.

[Saiba mais](#)

- HSETNX

Só define o valor de um campo em um hash quando o campo não existe.

[Saiba mais](#)

- HSTRLEN

Retorna o tamanho do valor de um campo.

[Saiba mais](#)

- HVALS

Retorna todos os valores em um hash.

[Saiba mais](#)

Comandos HyperLogLog

- PFADD

Adiciona elementos a uma chave do HyperLogLog. Cria a chave, caso ela não exista.

[Saiba mais](#)

- PFCOUNT

Retorna a cardinalidade aproximada dos conjuntos observados pelas chaves do HyperLogLog.

[Saiba mais](#)

- PFMERGE

Mescla um ou mais valores do HyperLogLog 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

Extraí um elemento de uma lista, o envia para outra lista e o retorna. Bloqueia até que um elemento esteja disponível de outra forma. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- LINDEX

Retorna um elemento de uma lista pelo índice.

[Saiba mais](#)

- LINSERT

Insere um elemento antes ou depois de outro elemento em uma lista.

[Saiba mais](#)

- LLEN

Retorna o tamanho de uma lista.

[Saiba mais](#)

- LMOVE

Retorna um elemento depois de extraí-lo de uma lista e enviá-lo para outra. Exclui a lista se o último elemento tiver sido movido.

[Saiba mais](#)

- LMPOP

Retornará vários elementos de uma lista depois de removê-los. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- LPOP

Retornará os primeiros elementos em uma lista depois de removê-la. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- LPOS

Retorna o índice de elementos correspondentes em uma lista.

[Saiba mais](#)

- LPUSH

Anexa um ou mais elementos a uma lista. Cria a chave, caso ela não exista.

[Saiba mais](#)

- LPUSHX

Só anexa um ou mais elementos a uma lista quando a lista existe.

[Saiba mais](#)

- LRANGE

Retorna um intervalo de elementos de uma lista.

[Saiba mais](#)

- LREM

Remove elementos de uma lista. Exclui a lista se o último elemento tiver sido removido.

[Saiba mais](#)

- LSET

Define o valor de um elemento em uma lista pelo índice.

[Saiba mais](#)

- LTRIM

Remove elementos de ambas as extremidades de uma lista. Exclui a lista se todos os elementos tiverem sido cortados.

[Saiba mais](#)

- RPOP

Retorna e remove os últimos elementos de uma lista. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- RPOPLPUSH

Retorna o último elemento de uma lista depois de removê-lo e enviá-lo para outra lista. Exclui a lista se o último elemento tiver sido extraído.

[Saiba mais](#)

- R PUSH

Acrescenta um ou mais elementos a uma lista. Cria a chave, caso ela não exista.

[Saiba mais](#)

- R PUSHX

Acrescenta um elemento a uma lista somente quando a lista existe.

[Saiba mais](#)

Comandos Pub/Sub

Note

Os comandos PUBSUB usam internamente o PUBSUB fragmentado, logo, os nomes de canais serão misturados.

- PUBLISH

Publica uma mensagem em um canal.

[Saiba mais](#)

- PUBSUB CHANNELS

Retorna os canais ativos.

[Saiba mais](#)

- PUBSUB NUMSUB

Retorna uma contagem de assinantes nos canais.

[Saiba mais](#)

- PUBSUB SHARDCHANNELS

Retorna os canais de fragmento ativos.

[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.

Comandos de script

- EVAL

Executa um script Lua do lado do servidor.

[Saiba mais](#)

- EVAL_RO

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_RO

Executa um script Lua do lado do servidor somente leitura por resumo 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 categorias ACL ou os comandos dentro de uma categoria.

[Saiba mais](#)

- ACL GENPASS

Gera uma senha pseudoaleatória e segura que pode ser usada para identificar usuários ACL.

[Saiba mais](#)

- ACL GETUSER

Lista as regras ACL de um usuário.

[Saiba mais](#)

- ACL LIST

Despeja as regras efetivas em formato de arquivo ACL.

[Saiba mais](#)

- ACL USERS

Lista todos os usuários ACL.

[Saiba mais](#)

- ACL WHOAMI

Retorna o nome de usuário autenticado da conexão atual.

[Saiba mais](#)

- DBSIZE

Retorna o número de chaves no banco de dados selecionado no momento. Não é garantido que essa operação seja atômica em todos os slots.

[Saiba mais](#)

- COMMAND

[Retorna informações detalhadas sobre todos os comandos.](#)

[Saiba mais](#)

- `COMMAND COUNT`

Retorna uma contagem de comandos.

[Saiba mais](#)

- `COMMAND DOCS`

Retorna informações documentais sobre um, vários ou todos os comandos.

[Saiba mais](#)

- `COMMAND GETKEYS`

Extrai os nomes de chaves de um comando arbitrário.

[Saiba mais](#)

- `COMMAND GETKEYSANDFLAGS`

Extrai os nomes das chaves e os sinalizadores de acesso de um comando arbitrário.

[Saiba mais](#)

- `COMMAND INFO`

Retorna informações sobre um, vários ou todos os comandos.

[Saiba mais](#)

- `COMMAND LIST`

Retorna uma lista de nomes de comando.

[Saiba mais](#)

- `FLUSHALL`

Remove todas as chaves de todos os bancos de dados. Não é garantido que essa operação seja atômica em todos os slots.

[Saiba mais](#)

- `FLUSHDB`

Remove todas as chaves do banco de dados atual. Não é garantido que essa operação seja atômica em todos os slots.

[Saiba mais](#)

- INFO

Retorna informações e estatísticas sobre o servidor.

[Saiba mais](#)

- LOLWUT

Exibe arte por computador e a versão do Redis.

[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.

[Saiba mais](#)

- SPOP

Retorna um ou mais membros aleatórios de um conjunto depois de removê-los. Exclui o conjunto se o último membro tiver sido extraído.

[Saiba mais](#)

- SRANDMEMBER

Obtenha um ou vários membros aleatórios de um conjunto

[Saiba mais](#)

- SREM

Remove um ou mais membros de um conjunto. Exclui o conjunto se o último membro tiver sido removido.

[Saiba mais](#)

- SSCAN

Itera membros de um conjunto.

[Saiba mais](#)

- SUNION

Retorna a união de vários conjuntos.

[Saiba mais](#)

- SUNIONSTORE

Armazena a união de vários conjuntos em uma chave.

[Saiba mais](#)

Comandos de conjunto classificado

- BZMPOP

Remove e retorna um membro por pontuação de um ou mais conjuntos classificados. Bloqueia até que um membro esteja disponível de outra forma. Exclui o conjunto classificado se o último elemento tiver sido extraído.

[Saiba mais](#)

- BZPOPMAX

Remove e retorna o membro com a pontuação mais alta de um ou mais conjuntos classificados. Bloqueia até que um membro esteja disponível de outra forma. Exclui o conjunto classificado se o último elemento tiver sido extraído.

[Saiba mais](#)

- BZPOPMIN

Remove e retorna o membro com a pontuação mais baixa de um ou mais conjuntos classificados. Bloqueia até que um membro esteja disponível de outra forma. Exclui o conjunto classificado se o último elemento tiver sido extraído.

[Saiba mais](#)

- ZADD

Adiciona um ou mais membros a um conjunto classificado ou atualiza as pontuações. Cria a chave, caso ela não exista.

[Saiba mais](#)

- ZCARD

Retorna o número de membros em um conjunto classificado.

[Saiba mais](#)

- ZCOUNT

Retorna a contagem de membros em um conjunto classificado que tenham pontuações dentro de um intervalo.

[Saiba mais](#)

- ZDIFF

Retorna a diferença entre vários conjuntos classificados.

[Saiba mais](#)

- ZDIFFSTORE

Armazena a diferença de vários conjuntos classificados em uma chave.

[Saiba mais](#)

- ZINCRBY

Incrementa a pontuação de um membro em um conjunto classificado.

[Saiba mais](#)

- ZINTER

Retorna a interseção de vários conjuntos classificados.

[Saiba mais](#)

- ZINTERCARD

Retorna o número de membros da interseção de vários conjuntos classificados.

[Saiba mais](#)

- ZINTERSTORE

Armazena a interseção de vários conjuntos classificados em uma chave.

[Saiba mais](#)

- ZLEXCOUNT

Retorna o número de membros em um conjunto classificado dentro de um intervalo lexicográfico.

[Saiba mais](#)

- ZMPOP

Retorna os membros com maior ou menor pontuação de um ou mais conjuntos classificados depois de removê-los. Exclui o conjunto classificado se o último membro tiver sido extraído.

[Saiba mais](#)

- ZMSCORE

Retorna a pontuação de um ou mais membros em um conjunto classificado.

[Saiba mais](#)

- ZPOPMAX

Retorna os membros com maior pontuação de um conjunto classificado depois de removê-los. Exclui o conjunto classificado se o último membro tiver sido extraído.

[Saiba mais](#)

- ZPOPMIN

Retorna os membros com menor pontuação de um conjunto classificado depois de removê-los. Exclui o conjunto classificado se o último membro tiver sido extraído.

[Saiba mais](#)

- ZRANDMEMBER

Retorna um ou mais membros aleatórios de um conjunto classificado.

[Saiba mais](#)

- ZRANGE

Retorna membros em um conjunto classificado dentro de um intervalo de índices.

[Saiba mais](#)

- ZRANGEBYLEX

Retorna membros em um conjunto classificado dentro de um intervalo lexicográfico.

[Saiba mais](#)

- ZRANGEBYSCORE

Retorna membros em um conjunto classificado dentro de um intervalo de índices.

[Saiba mais](#)

- ZRANGESTORE

Armazena um intervalo de membros do conjunto classificado em uma chave.

[Saiba mais](#)

- ZRANK

Retorna o índice de um membro em um conjunto classificado ordenado por pontuações crescentes.

[Saiba mais](#)

- ZREM

Remove um ou mais membros de um conjunto classificado. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREMRANGEBYLEX

Remove membros em um conjunto classificado dentro de um intervalo lexicográfico. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREMRANGEBYRANK

Remove membros em um conjunto classificado dentro de um intervalo de índices. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREMRANGEBYSCORE

Remove membros em um conjunto classificado dentro de um intervalo de pontuações. Exclui o conjunto classificado se todos os membros tiverem sido removidos.

[Saiba mais](#)

- ZREVRANGE

Retorna membros em um conjunto classificado dentro de um intervalo de índices em ordem inversa.

[Saiba mais](#)

- ZREVRANGEBYLEX

Retorna membros em um conjunto classificado dentro de um intervalo lexicográfico em ordem inversa.

[Saiba mais](#)

- ZREVRANGEBYSCORE

Retorna membros em um conjunto classificado dentro de um intervalo de classificações em ordem inversa.

[Saiba mais](#)

- ZREVRANK

Retorna o índice de um membro em um conjunto classificado ordenado por pontuações decrescentes.

[Saiba mais](#)

- ZSCAN

Itera membros e pontuações de um conjunto classificado.

[Saiba mais](#)

- ZSCORE

Retorna a pontuação de um membro em um conjunto classificado.

[Saiba mais](#)

- ZUNION

Retorna a união de vários conjuntos classificados.

[Saiba mais](#)

- ZUNIONSTORE

Armazena a união de vários conjuntos classificados em uma chave.

[Saiba mais](#)

Comandos de fluxo

- XACK

Retorna o número de mensagens que foram confirmadas com êxito pelo membro do grupo de consumidores de um fluxo.

[Saiba mais](#)

- XADD

Acrescenta uma nova mensagem a um fluxo. Cria a chave, caso ela não exista.

[Saiba mais](#)

- XAUTOCLAIM

Altera, ou adquire, a propriedade de mensagens em um grupo de consumidores, como se as mensagens tivessem sido entregues como membro do grupo de consumidores.

[Saiba mais](#)

- XCLAIM

Altera, ou adquire, a propriedade de uma mensagem em um grupo de consumidores, como se a mensagem tivesse sido entregue como membro do grupo de consumidores.

[Saiba mais](#)

- XDEL

Retorna o número de mensagens depois de removê-las de um fluxo.

[Saiba mais](#)

- XGROUP CREATE

Cria um grupo de consumidores.

[Saiba mais](#)

- XGROUP CREATECONSUMER

Cria um consumidor em um grupo de consumidores.

[Saiba mais](#)

- ~~XGROUP DELCONSUMER~~

Exclui um consumidor de um grupo de consumidores.

[Saiba mais](#)

- XGROUP DESTROY

Destrói um grupo de consumidores.

[Saiba mais](#)

- XGROUP SETID

Define o ID entregue mais recentemente de um grupo de consumidores.

[Saiba mais](#)

- XINFO CONSUMERS

Retorna uma lista dos consumidores em um grupo de consumidores.

[Saiba mais](#)

- XINFO GROUPS

Retorna uma lista dos grupos de consumidores de um fluxo.

[Saiba mais](#)

- XINFO STREAM

Retorna informações sobre um fluxo.

[Saiba mais](#)

- XLEN

Retorna o número de mensagens em um fluxo.

[Saiba mais](#)

- XPENDING

Retorna as informações e as entradas da lista de entradas pendentes de um grupo de consumidores do fluxo.

[Saiba mais](#)

- XRANGE

Retorna as mensagens de um fluxo dentro de um intervalo de IDs.

[Saiba mais](#)

- XREAD

Retorna mensagens de vários fluxos com IDs maiores do que os solicitados. 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 em 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 atômicamente os valores de string de uma ou mais chaves.

[Saiba mais](#)

- MSET

Cria ou modifica atômicamente os valores de string de uma ou mais chaves.

[Saiba mais](#)

- MSETNX

Só modifica atômicamente 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 Redis

Para fornecer uma experiência de serviço gerenciado, o ElastiCache restringe o acesso a certos 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 getredirect`
- `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`
- `punsubscribe`

- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`
- `wait`
- `watch`

Configuração e limites do Redis

O mecanismo Redis oferece vários parâmetros de configuração, alguns dos quais são modificáveis no ElastiCache para Redis e outros não são modificáveis para proporcionar desempenho estável e confiabilidade.

Caches sem servidor

Para caches sem servidor, grupos de parâmetros não são usados e todas as configurações do Redis não são modificáveis. Os seguintes parâmetros do Redis estão em vigor:

Nome	Detalhes	Descrição
<code>acl-pubsub-default</code>	<code>allchannels</code>	Permissões de canal pubsub padrão para usuários ACL no cache.
<code>client-output-buffer-limit</code>	<code>normal 0 0 0</code> <code>pubsub 32mb 8mb 60</code>	Cientes normais não têm limite de buffer. Os clientes PUB/SUB serão desconectados se violarem o backlog de 32 MiB ou o backlog de 8 MiB por 60 s.

Nome	Detalhes	Descrição
<code>client-query-buffer-limit</code>	15 GiB	O tamanho máximo de um único buffer de consulta do cliente. Além disso, os clientes não podem emitir uma solicitação com mais de 4.000 argumentos.
<code>cluster-allow-pubsubshard-when-down</code>	yes	Isso permite que o cache ofereça tráfego pubsub enquanto o cache está parcialmente inativo.
<code>cluster-allow-reads-when-down</code>	yes	Isso permite que o cache ofereça tráfego de leitura enquanto o cache está parcialmente inativo.
<code>cluster-enabled</code>	yes	Todos os caches sem servidor estão habilitados para o modo de cluster, o que permite particionar os dados de maneira transparente em vários fragmentos de back-end. Todos os slots são apresentados aos clientes como pertencentes a um único nó virtual.
<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> .

Nome	Detalhes	Descrição
<code>lua-time-limit</code>	5000	<p>O tempo de execução máximo para um script Lua, em milissegundos, antes que o ElastiCache tome medidas para interromper o script.</p> <p>Se <code>lua-time-limit</code> for excedido, todos os comandos do Redis poderão retornar um erro no formato <code>____-BUSY</code>. Como esse estado pode causar interferência com muitas operações essenciais do Redis, o ElastiCache primeiro emite um comando <code>SCRIPT KILL</code>. Se isso não tiver êxito, o ElastiCache reinicia o Redis de forma forçada.</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.
<code>maxmemory-policy</code>	<code>volatile-lru</code>	Os itens com um TTL definidos são removidos de acordo com a estimativa de uso menos recente (LRU) quando o limite de memória de um cache é atingido.
<code>notify-keyspace-events</code>	(uma string vazia)	No momento, os eventos do espaço de chaves não são compatíveis com caches sem servidor.
<code>port</code>	Porta primária: 6379 Porta de leitura: 6380	Os caches sem servidor anunciam duas portas com o mesmo nome de host. A porta primária permite gravações e leituras, e a porta de leitura permite leituras finais consistentes de latência mais baixa usando o comando <code>READONLY</code> .

Nome	Detalhes	Descrição
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 do Redis ou nome de canal. Os clientes que fizerem referência a chaves maiores do que isso receberão um erro.
Tamanho do script Lua	50 MiB	O tamanho máximo de um único script Lua do Redis. 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 hash do Redis. Os clientes que tentarem definir mais dados do que isso em um único slot do Redis vão disparar a política de remoção no slot e, se nenhuma chave for removível, receberão um erro de falta de memória (OOM).

Clusters autoprotetidos

Para clusters autoprotetidos, consulte [Parâmetros específicos do Redis](#) 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.

Práticas recomendadas com clientes do Redis

Conheça as práticas recomendadas para cenários comuns e acompanhe exemplos de código de algumas das bibliotecas cliente Redis de código-fonte aberto mais conhecidas (redis-py, phpRedis e Lettuce).

Tópicos

- [Número de conexões grande](#)
- [Descoberta de cliente do cluster do Redis e recuo exponencial](#)
- [Configurar um tempo limite do lado do cliente](#)
- [Configurar um tempo limite de inatividade do lado do servidor](#)
- [Scripts Lua do Redis](#)
- [Armazenamento de itens compostos grandes](#)
- [Configuração do cliente do Lettuce](#)
- [Exemplos de cliente de IPv6](#)

Número de conexões grande

Caches sem servidor e nós do ElastiCache para Redis individuais dão suporte a até 65 mil conexões de cliente simultâneas. No entanto, para otimizar o desempenho, recomendamos que as aplicações cliente não funcionem de maneira constante nesse nível de conexões. O Redis é um processo de thread único baseado em um loop de eventos no qual as solicitações de cliente recebidas são processadas sequencialmente. Isso significa que o tempo de resposta de um determinado cliente aumenta à medida que o número de clientes conectados aumenta.

É possível realizar o seguinte conjunto de ações para evitar um gargalo de conexões no servidor do Redis:

- Realize operações de leitura a partir das réplicas de leitura. Isso pode ser feito usando-se os endpoints de leitor do ElastiCache em modo cluster desabilitado ou usando réplicas para leituras em modo de cluster habilitado, inclusive um cache sem servidor.
- Distribua tráfego de gravação em vários nós primários. É possível fazer isso de duas maneiras. É possível usar um cluster do Redis multifragmentado com um cliente compatível com o

modo de cluster do Redis. Também é possível gravar em vários nós primários em modo de cluster desabilitado com fragmentação do lado do cliente. Isso é feito automaticamente em um cache sem servidor.

- Use um pool de conexões quando disponível na biblioteca cliente.

Em geral, criar uma conexão TCP é uma operação cara em termos computacionais em comparação com os comandos do Redis típicos. Por exemplo, lidar com uma solicitação SET/GET é uma ordem de magnitude mais rápida ao reutilizar uma conexão existente. O uso de um pool de conexões de clientes com um tamanho finito reduz a sobrecarga do gerenciamento de conexões. Isso também limita o número de conexões de entrada simultâneas da aplicação cliente.

O seguinte exemplo de código do PHPRedis mostra que uma nova conexão é criada para cada nova solicitação do usuário:

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
    //ERROR: connection failed
    return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Comparamos esse código em um loop em uma instância do Amazon Elastic Compute Cloud (Amazon EC2) conectada a um nó Graviton2 (m6g.2xlarge) do ElastiCache para o Redis. Colocamos o cliente e o servidor na mesma zona de disponibilidade. A latência média de toda a operação foi de 2,82 milissegundos.

Quando atualizamos o código e usamos conexões persistentes e um pool de conexões, a latência média de toda a operação foi de 0,21 milissegundo:

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
    // ERROR: connection failed
    return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Configurações de redis.ini necessárias:

- `redis.pconnect.pooling_enabled=1`
- `redis.pconnect.connection_limit=10`

O seguinte código é um exemplo de um [pool de conexões do Redis-py](#):

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
conn.set(key, value)
```

O seguinte código é um exemplo de um [pool de conexões do Lettuce](#):

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
GenericObjectPool<StatefulRedisConnection> pool =
    ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new
    GenericObjectPoolConfig());
pool.setMaxTotal(10); // Configure max connections to 10
try (StatefulRedisConnection connection = pool.borrowObject()) {
    RedisCommands syncCommands = connection.sync();
    syncCommands.set(key, value);
}
```

Descoberta de cliente do cluster do Redis e recuo exponencial

Ao se conectar a um cluster do ElastiCache para Redis em modo de cluster habilitado, a biblioteca cliente do Redis correspondente deve estar ciente do cluster. Os clientes devem obter um mapa dos slots de hash para os nós correspondentes no cluster para enviar solicitações aos nós certos e evitar a sobrecarga de desempenho de processar redirecionamentos de cluster. Dessa forma, o cliente deve descobrir uma lista completa de slots e dos nós mapeados em duas situações diferentes:

- O cliente é inicializado e deve preencher a configuração de slots inicial
- Um redirecionamento MOVED é recebido do servidor, como na situação de um failover, quando todos os slots atendidos pelo nó primário anterior são assumidos pela réplica, ou de refragmentação quando slots são movidos do nó primário de origem para o nó primário de destino

A descoberta de cliente costuma ser feita por meio da emissão de um comando CLUSTER SLOT ou CLUSTER NODE para o servidor do Redis. Recomendamos o método CLUSTER SLOT porque ele

retorna o conjunto de intervalos de slots e os nós primários e de réplica reassociados ao cliente. Isso não exige análise adicional do cliente e é mais eficiente.

Dependendo da topologia do cluster, o tamanho da resposta para o comando `CLUSTER SLOT` pode variar com base no tamanho do cluster. Clusters maiores com mais nós produzem uma resposta maior. Assim, é importante garantir que o número de clientes que fazem a descoberta da topologia do cluster não cresça de maneira ilimitada. Por exemplo, quando a aplicação cliente é inicializada ou perde a conexão do servidor e precisa realizar a descoberta de cluster, um erro comum é que a aplicação cliente dispara várias solicitações de reconexão e descoberta sem adicionar um recuo exponencial ao tentar novamente. Isso pode fazer o servidor do Redis não responder por um período prolongado de tempo, com a utilização da CPU em 100%. A interrupção será prolongada se cada comando `CLUSTER SLOT` precisar processar um número grande de nós no barramento do cluster. Observamos várias interrupções de cliente por causa desse comportamento em várias linguagens diferentes, inclusive Python (`redis-py-cluster`) e Java (`Lettuce` e `Redisson`).

Em um cache sem servidor, muitos dos problemas são atenuados automaticamente porque a topologia do cluster anunciada é estática e consiste em duas entradas: um endpoint de gravação e um endpoint de leitura. A descoberta de cluster também é distribuída automaticamente por vários nós durante o uso do endpoint de cache. No entanto, as recomendações a seguir continuam sendo úteis.

Para mitigar o impacto causado por um fluxo repentino de solicitações de conexão e descoberta, recomendamos o seguinte:

- Implemente um pool de conexões do cliente com um tamanho finito para limitar o número de conexões de entrada simultâneas da aplicação cliente.
- Quando o cliente se desconectar do servidor por causa do tempo limite, tente novamente com recuo exponencial com instabilidade. Isso ajuda a evitar que vários clientes sobrecarreguem o servidor ao mesmo tempo.
- Use o guia em [Localização de endpoints de conexão](#) para encontrar o endpoint do cluster a fim de realizar a descoberta do cluster. Ao fazer isso, você distribui a carga de descoberta em todos os nós do cluster (até 90) em vez de atingir alguns nós propagados codificados no cluster.

Estes são alguns exemplos de código para a lógica de nova tentativa de recuo exponencial em `redis-py`, `phpRedis` e `Lettuce`.

Exemplo 1 da lógica de recuo: `redis-py`

O `redis-py` tem um mecanismo de nova tentativa integrado repetido uma vez logo depois de uma falha. Esse mecanismo pode ser habilitado por meio do argumento `retry_on_timeout` fornecido ao criar um objeto do [Redis](#). Aqui, demonstramos um mecanismo de nova tentativa personalizado com recuo exponencial e instabilidade. Enviamos uma solicitação pull para implementar nativamente o recuo exponencial em [redis-py \(#1494\)](#). No futuro, talvez não seja necessário implementar manualmente.

```
def run_with_backoff(function, retries=5):
    base_backoff = 0.1 # base 100ms backoff
    max_backoff = 10 # sleep for maximum 10 seconds
    tries = 0
    while True:
        try:
            return function()
        except (ConnectionError, TimeoutError):
            if tries >= retries:
                raise
            backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
            print(f"sleeping for {backoff:.2f}s")
            sleep(backoff)
            tries += 1
```

É possível acabar usando o seguinte código para definir um valor:

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

Dependendo da workload, talvez você queira alterar o valor de recuo base de 1 segundo para algumas dezenas ou centenas de milissegundos para workloads sensíveis à latência.

Exemplo 2 da lógica de recuo: `PHPRedis`

O `PHPRedis` tem um mecanismo de nova tentativa integrado que repete um máximo (não configurável) de dez vezes. Há um atraso configurável entre as tentativas (com uma instabilidade a partir da segunda tentativa). Para obter mais informações, consulte o [código de exemplo](#). Enviamos uma solicitação pull para implementar nativamente o recuo exponencial em [phpRedis \(#1986\)](#), que já foi mesclado e [documentado](#). Para aqueles que estão na versão mais recente do `PHPRedis`, não será necessário implementar manualmente, mas incluímos aqui a referência para aqueles

nas versões anteriores. Por enquanto, este é um exemplo de código que configura o atraso do mecanismo de nova tentativa:

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
    return; // ERROR: connection failed
}
$client->set($key, $value);
```

Exemplo 3 da lógica de recuo: Lettuce

O Lettuce tem mecanismos de nova tentativa integrados com base nas estratégias de recuo exponencial baseado na publicação [Recuo exponencial e instabilidade](#). Este é um trecho de código que mostra a abordagem de instabilidade completa:

```
public static void main(String[] args)
{
    ClientResources resources = null;
    RedisClient client = null;

    try {
        resources = DefaultClientResources.builder()
            .reconnectDelay(Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(5), // maximum 5 second delay
                100, TimeUnit.MILLISECONDS) // 100 millisecond base
            ).build();

        client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
        client.setOptions(ClientOptions.builder()
            .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
            100 millisecond connection timeout
            .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //
            5 second command timeout
            .build());

        // use the connection pool from above example
    } finally {
        if (connection != null) {
            connection.close();
        }
    }
}
```

```
if (client != null){
    client.shutdown();
}

if (resources != null){
    resources.shutdown();
}

}
}
```

Configurar um tempo limite do lado do cliente

Configure devidamente o tempo limite do lado do cliente a fim de permitir que o servidor tenha tempo suficiente para processar a solicitação e gerar a resposta. Isso também permitirá antecipar-se à falha se a conexão com o servidor não puder ser estabelecida. Determinados comandos do Redis podem ser mais caros do que outros em termos computacionais. Por exemplo, scripts Lua ou transações MULTI/EXEC que contenham vários comandos que devem ser executados de maneira atômica. Em geral, um tempo limite do lado do cliente maior é recomendado para evitar um tempo limite do cliente antes que a resposta seja recebida do servidor, inclusive o seguinte:

- Execução de comandos em várias teclas
- Execução de transações MULTI/EXEC ou scripts Lua que consistem em vários comandos do Redis individuais
- Leitura de valores grandes
- Realização de operações de bloqueio, como BLPOP

No caso de uma operação de bloqueio, como BLPOP, a prática recomendada é definir o tempo limite do comando como um número menor que o tempo limite do soquete.

Estes são exemplos de código para implementar um tempo limite do lado do cliente em redis-py, PHPRedis e Lettuce.

Exemplo 1 de configuração do tempo limite: redis-py

Este é um exemplo de código com redis-py:

```
# connect to Redis server with a 100 millisecond timeout
```

```
# give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
print(res)                       # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
                                     # less than the 2 second socket timeout
print(res)
```

Exemplo 2 de configuração do tempo limite: PHPRedis

Este é um exemplo de código com PHPRedis:

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
    return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n";                    // if there is a connection error

$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n";                    // less than the 2 second socket timeout
```

Exemplo 3 de configuração do tempo limite: Lettuce

Este é um exemplo de código com Lettuce:

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
    RedisClient client = null;
    StatefulRedisConnection<String, String> connection = null;
    try {
        client = RedisClient.create(RedisURI.create(HOST, PORT));
```

```
client.setOptions(ClientOptions.builder()
.socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
100 millisecond connection timeout
.timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
2 second command timeout
.build());

// use the connection pool from above example

commands.set("key", "value"); // will timeout after 2 seconds
commands.blpop(1, "list"); // 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

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 específicos do Redis](#). 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 do Redis

O Redis suporta mais de 200 comandos, inclusive aqueles para executar scripts Lua. No entanto, quando o assunto são scripts Lua, existem diversas armadilhas que podem afetar a memória e a disponibilidade do Redis.

Scripts Lua não parametrizados

Cada script Lua é armazenado em cache no servidor Redis antes de ser executado. Os scripts Lua não parametrizados são exclusivos, o que pode fazer com que o servidor Redis armazene um

número grande de scripts Lua e consome mais memória. Para mitigar isso, certifique-se de que todos os scripts Lua estejam parametrizados e execute regularmente o SCRIPT FLUSH para limpar scripts Lua em cache, se necessário.

O exemplo a seguir mostra como usar scripts parametrizados: Primeiro, temos um exemplo de uma abordagem não parametrizada que resulta em três scripts Lua em cache diferentes e não é recomendada:

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

Em vez disso, use o seguinte padrão para criar um único script capaz de aceitar parâmetros passados:

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

Scripts Lua de longa duração

Os scripts Lua podem executar vários comandos atomicamente, logo, podem demorar mais para serem concluídos do que um comando do Redis normal. Se o script Lua só executar operações somente leitura, você poderá pará-lo no meio. No entanto, assim que executa uma operação de gravação, o script Lua se torna imortal e deve ser executado até a conclusão. Um script Lua de longa execução que esteja sofrendo mutação pode fazer com que o servidor Redis deixe de responder por um longo tempo. Para mitigar esse problema, evite scripts Lua de longa duração e teste o script em um ambiente de pré-produção.

Script Lua com gravações furtivas

Existem algumas maneiras pelas quais um script Lua pode continuar gravando novos dados no Redis mesmo quando o Redis terminar `maxmemory`:

- O script começa quando o servidor Redis está abaixo de `maxmemory` e contém várias operações de gravação dentro
- O primeiro comando de gravação do script não está consumindo memória (como DEL), seguido de mais operações de gravação que consomem memória

- Você pode mitigar esse problema configurando uma política de remoção indicada no servidor Redis, exceto `noeviction`. Isso permite que o Redis remova itens e libere memória entre os scripts Lua.

Armazenamento de itens compostos grandes

Em alguns cenários, uma aplicação pode armazenar itens compostos grandes no Redis (como um conjunto de dados hash de vários GB). Não se trata de uma prática recomendada porque ela normalmente acarreta problemas de desempenho no Redis. Por exemplo, o cliente pode executar um comando `HGETALL` para recuperar toda a coleção hash de vários GB. Isso pode gerar uma pressão de memória significativa para o servidor Redis, armazenando em buffer o item grande no buffer de saída do cliente. Além disso, para migração de slots em modo de cluster, o ElastiCache não migra slots que contenham itens com tamanho serializado maior que 256 MB.

Para resolver os problemas de itens grandes, temos as seguintes recomendações:

- Divida o item composto grande em vários itens menores. Por exemplo, divida uma coleção de hash grande em campos de valor-chave individuais com um esquema de nome de chave que reflita da maneira indicada a coleção, como o uso de um prefixo comum no nome da chave para identificar a coleção de itens. Se precisar acessar vários campos na mesma coleção atômicamente, você poderá usar o comando `MGET` para recuperar vários valores-chave no mesmo comando.
- Se você tiver avaliado todas as opções e ainda não tiver conseguido dividir o conjunto de dados da coleção grande, tente usar comandos que operem em um subconjunto dos dados na coleção, e não em toda a coleção. Evite ter um caso de uso que exija que você recupere atômicamente toda a coleção de vários GB no mesmo comando. Um exemplo é usar os comandos `HGET` ou `HMGET` em vez de `HGETALL` em coleções de hash.

Configuração do cliente do Lettuce

Esta seção descreve as opções de configuração recomendadas de Java e Lettuce e como elas se aplicam aos clusters do ElastiCache.

As recomendações nesta seção foram testadas com o Lettuce versão 6.2.2.

Tópicos

- [Exemplo: configuração do Lettuce para modo de cluster e TLS habilitado](#)

- [Exemplo: configuração do Lettuce para modo de cluster desabilitado e TLS habilitado](#)

TTL de cache de DNS do Java

A JVM armazena em cache pesquisas de nome DNS. Ao resolver um nome de host para um endereço IP, a JVM armazena em cache o endereço IP para um período especificado, conhecido como Time-To-Live (TTL – Tempo de duração).

A escolha de um valor TTL envolve um equilíbrio entre latência e capacidade de resposta a mudança. Com TTLs mais curtos, os resolvedores de DNS percebem atualizações no DNS do cluster com maior rapidez. Isso pode fazer com que sua aplicação responda com maior rapidez às substituições ou a outros fluxos de trabalho pelos quais seu cluster passa. No entanto, se o TTL for muito baixo, ele aumentará o volume da consulta, o que pode aumentar a latência de sua aplicação. Embora não exista um valor TTL correto, vale a pena considerar o tempo que você pode esperar para que uma alteração seja implementada ao definir o TTL.

Como os nós do ElastiCache utilizam entradas de nome de DNS que podem mudar, recomendamos configurar a JVM com um valor TTL de cinco a dez segundos. Isso garante que, quando o endereço IP de um nó mudar, a aplicação poderá receber e usar o novo endereço IP do recurso consultando novamente a entrada DNS.

Em algumas configurações do Java, o TTL padrão da JVM é definido de maneira que jamais atualizará entradas DNS até a JVM ser reiniciada.

Para obter detalhes sobre como definir seu TTL da JVM, consulte [Como definir o TTL da JVM](#).

Versão do Lettuce

Recomendamos o Lettuce versão 6.2.2 ou posterior.

Endpoints

Quando você estiver usando clusters habilitados para o modo de cluster, defina o `redisUri` como o endpoint de configuração do cluster. A pesquisa de DNS para esse URI retorna uma lista de todos os nós disponíveis no cluster e é resolvida aleatoriamente para um deles durante a inicialização do cluster. Para obter mais detalhes sobre como a atualização de topologia funciona, consulte `dynamicRefreshResources` mais adiante neste tópico.

SocketOption

Habilite o [KeepAlive](#). Quando habilitada, essa opção reduz a necessidade de lidar com falhas nas conexões durante o runtime de comando.

Defina [Tempo limite de conexão](#) com base nos requisitos da aplicação e na workload. Para ter mais informações, consulte “Tempos limite” posteriormente neste tópico.

ClusterClientOption: opções do cliente habilitadas para o modo de cluster

Habilite [AutoReconnect](#) quando a conexão for perdida.

Defina [CommandTimeout](#). Para obter mais detalhes, consulte a seção “Tempos limite” mais adiante neste tópico.

Defina [nodeFilter](#) para filtrar os nós com falha da topologia. O Lettuce salva todos os nós encontrados na saída “nós de cluster” (como nós com o status PFAIL/FAIL) nas “partições” do cliente (também conhecidas como fragmentos). Durante o processo de criação da topologia do cluster, ele tenta se conectar a todos os nós da partição. Esse comportamento do Lettuce ao adicionar nós com falha pode causar erros de conexão (ou avisos) quando os nós são substituídos por algum motivo.

Por exemplo, depois que um failover é concluído e o cluster inicia o processo de recuperação, enquanto a topologia do cluster é atualizada, o mapa de nós de barramento do cluster tem um curto período em que o nó inativo é listado como um nó com o status FAIL antes de ser completamente removido da topologia. Durante esse período, o cliente do Lettuce Redis o considera um nó íntegro e se conecta de forma contínua a ele. Isso causa uma falha após o término da nova tentativa.

Por exemplo:

```
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        ... // other options
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
                || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
                || it.is(RedisClusterNode.NodeFlag.HANDSHAKE)
                || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();
redisClusterClient.setOptions(clusterClientOptions);
```

Note

A filtragem de nós é mais bem utilizada com `DynamicRefreshSources` definido como verdadeiro. Do contrário, se a visualização da topologia for obtida de um único nó inicial problemático, que vê um nó primário de algum fragmento com o status de falha, ela filtrará esse nó primário, o que causará a falta de cobertura dos slots. Ter vários nós iniciais (quando `DynamicRefreshSources` é verdadeiro) reduz a probabilidade de ocorrência desse problema, pois pelo menos alguns dos nós iniciais devem ter uma visualização de topologia atualizada após um failover com o primário recém-promovido.

`ClusterTopologyRefreshOptions`: opções para controlar a atualização da topologia do cluster do cliente Habilitado no modo do cluster

Note

Os clusters com modo de cluster desabilitado não dão suporte a comandos de descoberta do cluster nem são compatíveis com a funcionalidade de descoberta da topologia dinâmica de todos os clientes.

O modo de cluster desabilitado com o ElastiCache não é compatível com `MasterSlaveTopologyRefresh` do Lettuce. Em vez disso, para o modo de cluster desabilitado, você pode configurar um `StaticMasterReplicaTopologyProvider` e fornecer os endpoints de leitura e gravação do cluster.

Para obter mais informações sobre a conexão com clusters no modo de cluster desabilitado, consulte [Localização de endpoints de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#).

Se quiser usar a funcionalidade de descoberta da topologia dinâmica do Lettuce, você poderá criar um cluster com modo de cluster habilitado com a mesma configuração de fragmento do cluster existente. No entanto, para clusters habilitados no modo de cluster, recomendamos configurar pelo menos três fragmentos com no mínimo uma réplica para oferecer compatibilidade com failover rápido.

Habilite [enablePeriodicRefresh](#). Isso possibilita atualizações periódicas da topologia do cluster para que o cliente a atualize nos intervalos do `refreshPeriod` (padrão: 60 segundos). Quando desabilitado, o cliente atualiza a topologia do cluster somente quando ocorrem erros ao tentar executar comandos no cluster.

Com essa opção habilitada, você pode reduzir a latência associada à atualização da topologia do cluster adicionando esse trabalho a uma tarefa em segundo plano. Embora a atualização da topologia seja executada em um trabalho em segundo plano, ela pode ser um pouco lenta para clusters com muitos nós. Isso ocorre porque todos os nós estão sendo consultados para obter a visualização mais atualizada do cluster. Se você executa um cluster grande, é recomendável aumentar o período.

Habilite [enableAllAdaptiveRefreshTriggers](#). Isso possibilita a atualização adaptativa da topologia que usa todos os [acionadores](#): `MOVED_REDIRECT`, `ASK_REDIRECT`, `PERSISTENT_RECONNECTS`, `UNCOVERED_SLOT`, `UNKNOWN_NODE`. Os acionadores de atualização adaptável iniciam atualizações de visualização da topologia com base em eventos que ocorrem durante as operações de cluster do Redis. Quando habilitada, essa opção ocasiona uma atualização imediata da topologia quando ocorre um dos acionadores anteriores. As atualizações adaptáveis acionadas têm uma taxa limitada usando um tempo limite porque os eventos podem ocorrer em grande escala (tempo limite padrão entre as atualizações: 30).

Habilite [closeStaleConnections](#). Isso possibilita fechar conexões obsoletas ao atualizar a topologia do cluster. Ela só entrará em vigor se [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled\(\)](#) for verdadeiro. Quando habilitado, o cliente pode fechar conexões obsoletas e criar outras em segundo plano. Isso reduz a necessidade de lidar com falhas nas conexões durante o runtime de comando.

Habilite [dynamicRefreshResources](#). Recomendamos habilitar a opção `dynamicRefreshResources` para clusters pequenos e desabilitá-la para clusters grandes. A opção `dynamicRefreshResources` possibilita descobrir nós de cluster com base no nó inicial fornecido (por exemplo, endpoint de configuração de cluster). Ele usa todos os nós descobertos como fontes para atualizar a topologia do cluster.

A atualização dinâmica consulta todos os nós descobertos para a topologia do cluster e tenta escolher a visualização mais precisa do cluster. Se for definido como falso, somente os nós iniciais serão usados como fontes para a descoberta da topologia e o número de clientes será obtido apenas para os nós iniciais. Quando desabilitado, se o endpoint de configuração do cluster for resolvido em um nó com falha, a tentativa de atualizar a visualização do cluster falhará e causará exceções. Esse cenário pode ocorrer porque demora até que a entrada de um nó com falha seja removida do endpoint de configuração do cluster. Portanto, o endpoint de configuração ainda pode ser resolvido aleatoriamente em um nó com falha por um curto período.

No entanto, quando está habilitado, usamos todos os nós do cluster recebidos da visualização do cluster para consultar a visualização atual. Como filtramos os nós com falha nessa visualização, a atualização da topologia será bem-sucedida. No entanto, quando `dynamicRefreshSources` é

verdadeiro, o Lettuce consulta todos os nós para obter a visualização do cluster e, depois, compara os resultados. Portanto, isso pode ser caro para clusters com muitos nós. Sugerimos que você desabilite esse recurso para clusters com muitos nós.

```
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();
```

ClientResources

Configure [DnsResolver](#) com [DirContextDnsResolver](#). O resolvidor de DNS é baseado em `com.sun.jndi.dns.DnsContextFactory` do Java.

Configure [reconnectDelay](#) com recuo exponencial e oscilação total. O Lettuce tem mecanismos de repetição integrados com base nas estratégias de recuo exponencial. Para obter detalhes, consulte a publicação [Recuo exponencial e oscilação](#) no Blog de arquitetura da AWS. Para ter mais informações sobre a importância de tentar novamente uma estratégia de recuo, consulte as seções de lógica de recuo da [publicação do blog sobre práticas recomendadas](#) no Blog de bancos de dados da AWS.

```
ClientResources clientResources = DefaultClientResources.builder()
    .dnsResolver(new DirContextDnsResolver())
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100), // minimum 100 millisecond delay
            Duration.ofSeconds(10), // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .build();
```

Tempos limite

Use um valor de tempo limite de conexão menor do que o tempo limite do comando. O Lettuce utiliza uma conexão preguiçosa. Portanto, se o tempo limite de conexão for maior do que o tempo limite do comando, você poderá ter um período de falha persistente após uma atualização da topologia se o Lettuce tentar se conectar a um nó não íntegro e o tempo limite do comando sempre for excedido.

Utilize um tempo limite de comando dinâmico para comandos diferentes. Recomendamos definir o tempo limite do comando com base na duração esperada do comando. Por exemplo, utilize um

tempo limite maior para comandos que iteram em várias teclas, como scripts FLUSHDB, FLUSHALL, KEYS, SMEMBERS ou Lua. Utilize tempos limite mais curtos para comandos de tecla única, como SET, GET e HSET.

Note

Os tempos limite configurados no exemplo a seguir são para testes que executaram comandos SET/GET com chaves e valores de até 20 bytes. O tempo de processamento pode ser maior quando os comandos são complexos ou quando as chaves e os valores são maiores. Você deve definir os tempos limite com base no caso de uso de sua aplicação.

```
private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);
```

```
SocketOptions socketOptions = SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .build();
```

```
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
    ImmutableSet.<ProtocolKeyword>builder()
        .add(CommandType.FLUSHDB)
        .add(CommandType.FLUSHALL)
        .add(CommandType.CLUSTER)
        .add(CommandType.INFO)
        .add(CommandType.KEYS)
        .build();

    private final Duration defaultCommandTimeout;
    private final Duration metaCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
```

```
public long getTimeout(RedisCommand<?, ?, ?> command) {
    if (META_COMMAND_TYPES.contains(command.getType())) {
        return metaCommandTimeout.toMillis();
    }
    return defaultCommandTimeout.toMillis();
}
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(
        new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();
```

Exemplo: configuração do Lettuce para modo de cluster e TLS habilitado

Note

Os tempos limite no exemplo a seguir são para testes que executaram comandos SET/GET com chaves e valores de até 20 bytes. O tempo de processamento pode ser maior quando os comandos são complexos ou quando as chaves e os valores são maiores. Você deve definir os tempos limite com base no caso de uso de sua aplicação.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
    cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)
```

```
        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),    // minimum 100 millisecond delay
            Duration.ofSeconds(10),    // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
    ImmutableSet.<ProtocolKeyword>builder()
        .add(CommandType.FLUSHDB)
        .add(CommandType.FLUSHALL)
        .add(CommandType.CLUSTER)
        .add(CommandType.INFO)
        .add(CommandType.KEYS)
        .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}
```

```
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
META_COMMAND_TIMEOUT))
    .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();

// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
        .connectTimeout(CONNECT_TIMEOUT)
        .keepAlive(true)
        .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        .topologyRefreshOptions(topologyOptions)
        .socketOptions(socketOptions)
        .autoReconnect(true)
        .timeoutOptions(timeoutOptions)
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
            || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
            || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
```

```
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Exemplo: configuração do Lettuce para modo de cluster desabilitado e TLS habilitado

Note

Os tempos limite no exemplo a seguir são para testes que executaram comandos SET/GET com chaves e valores de até 20 bytes. O tempo de processamento pode ser maior quando os comandos são complexos ou quando as chaves e os valores são maiores. Você deve definir os tempos limite com base no caso de uso de sua aplicação.

```
// Set DNS cache TTL
public void setJVMPProperties() {
    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);
```

Exemplos de cliente de IPv6

A seguir estão as melhores práticas para interagir com ElastiCache recursos habilitados para IPv6 com bibliotecas cliente de código aberto comumente usadas. Você pode ver [as melhores práticas existentes para interagir ElastiCache](#) para obter recomendações sobre como configurar clientes para ElastiCache recursos. No entanto, há algumas ressalvas que vale a pena observar ao interagir com recursos habilitados para IPv6.

Cientes validados

ElastiCache é compatível com o Redis de código aberto. Isso significa que os clientes Redis de código aberto que oferecem suporte a conexões IPv6 devem ser capazes de se conectar ao IPv6 habilitado para clusters Redis. Além disso, vários dos clientes Python e Java mais populares foram especificamente testados e validados para funcionar com todas as configurações de tipo de rede compatíveis (somente IPv4, somente IPv6 e pilha dupla)

Cientes validados:

- [Redis Py \(\)](#) – [4.1.2](#)
- [Lettuce](#) – [Version: 6.1.6.RELEASE](#)
- [Jedis](#) – [Versão: 3.6.0](#)

Configurar um protocolo preferencial para clusters de pilha dupla

Para clusters do Redis habilitados para o modo cluster, você pode controlar o protocolo que os clientes usarão para se conectar aos nós no cluster com o parâmetro de descoberta de IP. O parâmetro de descoberta de IP pode ser definido como IPv4 ou IPv6.

Para clusters do Redis, o parâmetro de descoberta de IP define o protocolo IP usado na saída de [cluster slots \(\)](#), [cluster shards \(\)](#) e [cluster nodes \(\)](#). Esses comandos são usados pelos clientes para descobrir a topologia do cluster. Os clientes usam os IPs nesses comandos para se conectar aos outros nós no cluster.

Alterar a descoberta de IP não resultará em nenhum tempo de inatividade para os clientes conectados. No entanto, as alterações levarão algum tempo para se propagar. Para determinar quando as alterações foram completamente propagadas para um cluster do Redis, monitore a saída de `cluster slots`. Depois que todos os nós retornados pelo comando `cluster slots` relatam IPs com o novo protocolo, as alterações terminam de se propagar.

Exemplo com Redis-Py:

```

cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
nodes):
    nodes = set()

    # This refreshes the cluster topology and will discovery any node updates.
    # Under the hood it calls cluster slots
    cluster.nodes_manager.initialize()
    for node in cluster.get_nodes():
        nodes.add(node.host)
    self.logger.info(nodes)

time.sleep(1)

```

Exemplo com Lettuce:

```

RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
6379));

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

Set<String> nodes;

do {
    // Check for any changes in the cluster topology.
    // Under the hood this calls cluster slots
    clusterClient.refreshPartitions();
    Set<String> nodes = new HashSet<>();

    for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
        nodes.add(node.getUri().getHost());
    }

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
    } catch (UnknownHostException ignored) {}
    return false;
}));

```

```
});
```

Clusters de ElastiCache pilha dupla habilitados para TLS

Quando o TLS está habilitado para ElastiCache clusters, as funções de descoberta de cluster (`cluster slots`, `cluster shards`, `ecluster nodes`) retornam nomes de host em vez de IPs. Os nomes de host são então usados em vez de IPs para se conectar ao ElastiCache cluster e realizar um handshake TLS. Isso significa que os clientes não serão afetados pelo parâmetro de descoberta de IP. Para clusters habilitados para TLS, o parâmetro de descoberta de IP não tem efeito no protocolo IP preferencial. Em vez disso, o protocolo IP usado será determinado pelo protocolo IP que o cliente prefere ao resolver nomes de host DNS.

Cientes Java

Ao se conectar de um ambiente Java que oferece suporte a IPv4 e IPv6, o Java, por padrão, prefere IPv4 em vez de IPv6 para compatibilidade com versões anteriores. No entanto, a preferência do protocolo IP é configurável por meio dos argumentos da JVM. Para preferir IPv4, a JVM aceita `-Djava.net.preferIPv4Stack=true` e prefere o conjunto IPv6 - `Djava.net.preferIPv6Stack=true`. A configuração `-Djava.net.preferIPv4Stack=true` significa que a JVM não fará mais nenhuma conexão IPv6. Incluindo-os em outras aplicações que não sejam do Redis.

Preferências de nível de host

Em geral, se o runtime do cliente ou o cliente não fornecer opções de configuração para definir uma preferência de protocolo IP, ao executar a resolução de DNS, o protocolo IP dependerá da configuração do host. Por padrão, a maioria dos hosts prefere IPv6 em vez de IPv4, mas essa preferência pode ser configurada no nível do host. Isso afetará todas as solicitações de DNS desse host, não apenas aquelas para 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 para preferir IPv4 ao se conectar a um ElastiCache cluster, atualize a precedência do intervalo CIDR que abrange os IPs do cluster para estar acima da precedência das conexões IPv6 padrão. Por padrão, as conexões IPv6 têm uma

precedência de 40. Por exemplo, supondo que o cluster esteja localizado em uma sub-rede com o CIDR 172.31.0.0/16, a configuração abaixo faria com que os clientes preferissem conexões IPv4 a esse cluster.

```
label ::1/128      0
label ::/0        1
label 2002::/16   2
label ::/96       3
label ::ffff:0:0/96 4
label fec0::/10   5
label fc00::/7    6
label 2001:0::/32 7
label ::ffff:172.31.0.0/112 8
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# site-local IPv4 and IPv6 addresses a lookup for a global address would
# see the IPv6 be preferred. The result is a long delay because the
# site-local IPv6 addresses cannot be used while the IPv4 address is
# (at least for the foreseeable future) NATed. We also treat Teredo
# tunnels special.
#
# precedence <mask> <value>
# Add another rule to the RFC 3484 precedence table. See section 2.1
# and 10.3 in RFC 3484. The default is:
#
precedence ::1/128      50
precedence ::/0        40
precedence 2002::/16   30
precedence ::/96       20
precedence ::ffff:0:0/96 10
precedence ::ffff:172.31.0.0/112 100
```

Mais detalhes sobre `gai.conf` estão disponíveis na [página principal do Linux](#)

Hosts do Windows

O processo para hosts do Windows é semelhante. Para hosts do Windows, você pode executar `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`. Isso tem o mesmo resultado que modificar o arquivo `gai.conf` em hosts Linux.

Isso atualizará as políticas de preferência para preferir conexões IPv4 em vez de conexões IPv6 para o intervalo CIDR especificado. Por exemplo, supondo que o cluster esteja em uma sub-rede com o CIDR 172.31.0.0:0/16, a execução de `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` resultaria na seguinte tabela de precedência, o que faria com que os clientes preferissem IPv4 ao se conectarem ao cluster.

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...
```

```
Precedence Label Prefix
```

```
-----
100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```

Práticas recomendadas durante o trabalho com clusters autoprojeados

Esta seção só se aplica quando você opta por projetar os próprios clusters do Redis. É recomendável analisar e seguir essas práticas recomendadas.

Tópicos

- [Minimização do tempo de inatividade com multi-AZ](#)
- [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#)
- [Gerenciamento da memória reservada](#)
- [Redimensionamento de cluster on-line](#)
- [Minimização do tempo de inatividade durante a manutenção](#)

Minimização do tempo de inatividade com multi-AZ

Consulte [Minimização do tempo de inatividade no ElastiCache for Redis com o multi-AZ](#), para obter mais informações sobre o multi-AZ e a minimização do tempo de inatividade.

Garantia de que você tenha memória suficiente para criar um snapshot do Redis

Snapshots e sincronizações do Redis na versão 2.8.22 e posterior

O Redis 2.8.22 apresenta um processo de salvamento sem bifurcação que permite alocar mais da sua memória ao uso do seu aplicativo sem incorrer em um maior uso de permuta durante sincronizações e salvamentos. Para obter mais informações, consulte [Como a sincronização e o backup são implementados](#).

Snapshots e sincronizações do Redis antes da versão 2.8.22

Quando você trabalha com o ElastiCache for Redis, o Redis chama um comando de gravação em segundo plano em vários casos:

- Ao criar um snapshot para um backup.
- Ao sincronizar réplicas com o primário em um grupo de replicação.
- Ao habilitar o recurso de arquivo somente de acréscimo (AOF) para o Redis.
- Ao promover uma réplica para primário (o que causa uma sincronização entre o primário/réplica).

Sempre que o Redis executa um processo de gravação em segundo plano, você deve ter memória disponível suficiente para acomodar a sobrecarga de processos. A falta de memória suficiente disponível faz com que o processo falhe. Por isso, é importante escolher um tipo de instância de nó que tenha memória suficiente ao criar seu cluster Redis.

Processo de gravação em segundo plano e uso de memória

Sempre que um processo de gravação em segundo plano é chamado, o Redis bifurca seu processo (lembre-se de que o Redis é single-threaded). Uma bifurcação persiste seus dados no disco em um arquivo de snapshot `.rdb` do Redis. A outra bifurcação atende a todas as operações de leitura e gravação. Para garantir que seu snapshot seja um snapshot de ponto no tempo, todas as atualizações e adições de dados são gravadas em uma área de memória disponível separada da área de dados.

Enquanto você tiver memória suficiente disponível para registrar todas as operações de gravação durante a manutenção dos dados no disco, não haverá problemas de memória insuficiente. É provável que você experimente problemas de memória insuficiente se uma das seguintes condições for verdadeira:

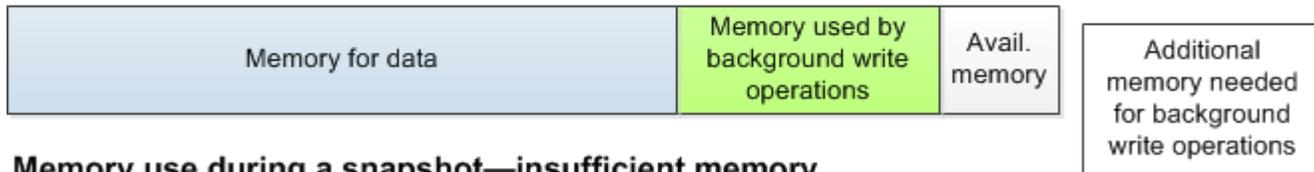
- Seu aplicativo realiza muitas operações de gravação, exigindo assim uma grande quantidade de memória disponível para aceitar os dados novos ou atualizados.
- Você tem pouca memória disponível para gravar dados novos ou atualizados.
- Você tem um grande conjunto de dados que demora muito para persistir no disco, exigindo um grande número de operações de gravação.

O diagrama a seguir ilustra o uso da memória ao executar um processo de gravação em segundo plano.

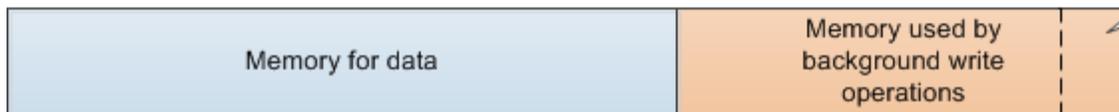
Memory use prior to a snapshot



Memory use during a snapshot—sufficient memory



Memory use during a snapshot—insufficient memory



Para obter informações sobre o impacto de fazer um backup sobre o desempenho, consulte [Impacto sobre o desempenho dos backups de clusters autoprojeto](#)dos.

Para obter mais informações sobre como o Redis realiza snapshots, consulte <http://redis.io>.

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Escolher regiões e zonas de disponibilidade](#).

Evitando memória insuficiente ao executar uma gravação em segundo plano

Sempre que um processo de gravação em segundo plano, como BGSAVE ou BGREWRITEAOF, é chamado, para evitar que o processo falhe, você deve ter mais memória disponível do que será consumida por operações de gravação durante o processo. O cenário de pior caso é aquele em que, durante a operação de gravação em segundo plano, cada registro do Redis é atualizado e alguns novos registros são adicionados ao cache. Por conta disso, recomendamos que você defina `reserved-memory-percent` como 50 (50 por cento) para versões do Redis anteriores à 2.8.22 ou como 25 (25 por cento) para versões do Redis 2.8.22 e posteriores.

O valor `maxmemory` indica a memória disponível para dados e sobrecargas operacionais. Como você não pode modificar o parâmetro `reserved-memory` no parameter group padrão, deve criar um parameter group personalizado para o cluster. O valor padrão para `reserved-memory` 0, o que permite que o Redis consuma todo o `maxmemory` com dados, potencialmente deixando pouca memória para outros usos, como um processo de gravação em segundo plano. Para valores `maxmemory` por tipo de instância de nó, consulte [Parâmetros específicos de node-type do Redis](#).

Você também pode usar o parâmetro `reserved-memory` para reduzir a quantidade de memória que o Redis utiliza na caixa.

Para obter mais informações sobre parâmetros específicos do Redis no ElastiCache, consulte [Parâmetros específicos do Redis](#).

Para obter informações sobre como criar e modificar parameter groups, consulte [Criar um parameter group](#) e [Modificar um parameter group](#).

Gerenciamento da memória reservada

A memória reservada é a memória reservada para o uso não relacionado a dados. Ao executar um backup ou failover, o Redis usa a memória disponível para realizar operações de gravação no seu cluster enquanto os dados do cluster estão sendo gravados no arquivo .rdb. Se você não tiver memória suficiente disponível para todas as gravações, o processo falhará. A seguir, você encontrará informações sobre opções para gerenciar memória reservada para o ElastiCache for Redis 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 anterior à 2.8.22, reserve mais memória para backups e failover do que se estiver executando o Redis 2.8.22 ou posterior. Este requisito é devido às diferentes maneiras como o ElastiCache for Redis implementa o processo de backup. A regra de ouro é reservar metade do valor `maxmemory` de um tipo de nó para a sobrecarga do Redis para versões anteriores a 2.8.22 e um quarto para as versões Redis 2.8.22 e posterior.

Ao usar clusters com classificação de dados em níveis, recomendamos aumentar o parâmetro `maxmemory` para até metade da memória disponível do nó se a workload tiver um uso intenso de gravação.

Para obter mais informações, consulte as informações a seguir.

- [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#)
- [Como a sincronização e o backup são implementados](#)
- [Classificação de dados em níveis](#)

Parâmetros para gerenciar memória reservada

Desde 16 de março de 2017, o Amazon ElastiCache for Redis fornece dois parâmetros mutuamente exclusivos para gerenciar sua memória do Redis, `reserved-memory` e `reserved-memory-percent`. Nenhum desses parâmetros faz parte da distribuição do Redis.

Dependendo de quando você se tornou cliente do ElastiCache, um desses dois parâmetros será o parâmetro de gerenciamento de memória padrão. Esse parâmetro se aplica quando você cria um novo cluster do Redis ou grupo de replicação e usa um grupo de parâmetros padrão.

- Para clientes que iniciaram antes de 16 de março de 2017, quando você criar um cluster Redis ou um grupo de replicação usando o parameter group padrão, o parâmetro de gerenciamento de memória será `reserved-memory`. Nesse caso, zero (0) bytes de memória são reservados.
- Para clientes que iniciaram de 16 de março de 2017 em diante, quando você criar um cluster Redis ou um grupo de replicação usando o parameter group padrão, o parâmetro de gerenciamento de memória será `reserved-memory-percent`. Nesse caso, 25% do valor `maxmemory` dos seus nós será reservado para fins não de dados.

Depois de ler sobre os dois parâmetros de gerenciamento de memória de Redis, você pode preferir usar aquele que não é o padrão, ou com valores não padrão. Se esse for o caso, você poderá alterar para o outro parâmetro de gerenciamento de memória reservada.

Para alterar o valor desse parâmetro, você pode criar um grupo de parâmetros personalizado e modificá-lo para usar o parâmetro e o valor de gerenciamento de memória preferidos. Você pode usar o parameter group personalizado sempre que criar um novo cluster ou grupo de replicação 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](#)
- [Criar um parameter group](#)
- [Modificar um parameter group](#)
- [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 reservado do ElastiCache for Redis era feito com o uso do parâmetro `reserved-memory`. O valor padrão de `reserved-memory` é 0. Este padrão não reserva memória para a sobrecarga da Redis e permite que o Redis consuma toda a memória de um nó com dados.

A alteração do parâmetro `reserved-memory` para que você tenha memória suficiente disponível para backups e failovers requer que você crie um `parameter group` personalizado. Neste grupo de parâmetros personalizado, você define `reserved-memory` como um valor apropriado para a versão do Redis executada no seu cluster e o tipo de nó do cluster. Para obter mais informações, consulte [De quanta memória reservada você precisa?](#)

O parâmetro `reserved-memory` do ElastiCache for Redis é específico para o ElastiCache for Redis e não faz parte da distribuição do Redis.

O procedimento a seguir mostra como usar `reserved-memory` para gerenciar a memória no seu cluster Redis.

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 [Criar um parameter group](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis6x-m3x1 \  
  --description "Redis 2.8.x for m3.xlarge node type" \  
  --cache-parameter-group-family redis6.x
```

2. Calcule quantos bytes de memória a reservar para a sobrecarga do Redis. Você pode encontrar o valor de `maxmemory` para o seu tipo de nó em [Parâmetros específicos de node-type do 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 seguinte exemplo de AWS CLI supõe que você esteja executando uma versão do Redis anterior à 2.8.22 e precisa reservar metade da `maxmemory` do nó. Para obter mais informações, consulte [Modificar um parameter group](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory,  
  ParameterValue=7130316800"
```

Observe que você precisa de um grupo de parâmetros personalizado separado para cada tipo de nó utilizado, pois cada tipo de nó possui um valor de `maxmemory` diferente. Assim, cada tipo de nó precisa de um valor diferente para `reserved-memory`.

4. Modifique seu cluster ou grupo de replicação Redis para usar seu parameter group personalizado.

O exemplo de CLI a seguir modifica o cluster `my-redis-cluster` para começar a usar imediatamente o parameter group `redis28-m3x1`. Para obter mais informações, consulte [Modificando um cluster ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

O exemplo de CLI a seguir modifica o grupo de replicação `my-redis-repl-grp` para começar a usar imediatamente o parameter group `redis28-m3x1`. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

O parâmetro `reserved-memory-percent`

Em 16 de março de 2017, o Amazon ElastiCache introduziu o parâmetro `reserved-memory-percent` e o disponibilizou em todas as versões do ElastiCache for Redis. 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` do ElastiCache for Redis é específico para o ElastiCache for Redis e não faz parte da distribuição 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 [Classificação de dados em níveis](#).

Para reservar memória usando `reserved-memory-percent`

Para usar `reserved-memory-percent` para gerenciar a memória no seu cluster do ElastiCache for Redis, siga um destes procedimentos:

- Se você estiver executando o Redis 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 anterior à 2.8.22, provavelmente precisará reservar mais memória do que o padrão de 25 por cento de `reserved-memory-percent`. Para fazer isso, use o procedimento a seguir.

Para alterar o valor percentual de `reserved-memory-percent`

1. Crie um `parameter group` personalizado especificando a família de `parameter groups` que corresponde à versão do mecanismo que você está executando, por exemplo, especificando a família de `parameter groups` `redis2.8`. Um grupo de parâmetros personalizado é necessário, pois você não pode modificar um grupo de parâmetros padrão. Para obter mais informações, consulte [Criar um parameter group](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --description "Redis 2.8.x 50% reserved" \  
  --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 [Modificar um parameter group](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --parameter-name-values "ParameterName=reserved-memory-percent,  
  ParameterValue=50"
```

3. Use esse `parameter group` personalizado para qualquer cluster ou grupo de replicação Redis que esteja executando uma versão do Redis anterior à 2.8.22.

O exemplo de CLI a seguir modifica o cluster Redis `my-redis-cluster` para começar a usar imediatamente o parameter group `redis28-50`. 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 exemplo de CLI a seguir modifica o grupo de replicação Redis `my-redis-repl-grp` para começar a usar imediatamente o parameter group `redis28-50`. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

Especificação do parâmetro de gerenciamento de memória reservada

Se você já era cliente ativo do ElastiCache em 16 de março de 2017, seu parâmetro de gerenciamento de memória reservada padrão é `reserved-memory` com zero (0) byte de memória reservada. Se você se tornou cliente do ElastiCache depois de 16 de março de 2017, seu parâmetro de gerenciamento de memória reservada padrão é `reserved-memory-percent`, com 25 por cento da memória do nó reservada. Isso se aplica sempre que você cria um cluster ou grupo de replicação do ElastiCache for Redis. Entretanto, você pode alterar seu parâmetro de gerenciamento de memória reservada usando a AWS CLI ou a API do ElastiCache.

Os parâmetros `reserved-memory` e `reserved-memory-percent` são mutuamente exclusivos. Um grupo de parâmetros sempre terá um, mas nunca os dois. Você pode alterar o parâmetro usado por um parameter group para o gerenciamento da memória reservada, modificando o parameter group. O grupo de parâmetros deve ser personalizado, pois não é possível modificar grupos de parâmetros padrão. Para obter mais informações, consulte [Criar um parameter group](#).

Para especificar `reserved-memory-percent`

Para usar `reserved-memory-percent` como o parâmetro de gerenciamento de memória reservada, modifique um grupo de parâmetros usando o comando `modify-cache-parameter-`

group. Use o parâmetro `parameter-name-values` para especificar `reserved-memory-percent` e um valor para ele.

O seguinte exemplo de CLI modifica o `parameter group` personalizado `redis32-cluster-on` para que ele use `reserved-memory-percent` para gerenciar a memória reservada. Um valor deve ser atribuído a `ParameterValue` para que o grupo de parâmetros use o parâmetro `ParameterName` para gerenciamento de memória reservada. Para obter mais informações, consulte [Modificar um parameter group](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-cluster-on \  
  --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

Para especificar reserved-memory

Para usar `reserved-memory` como o parâmetro de gerenciamento de memória reservada, modifique um grupo de parâmetros usando o comando `modify-cache-parameter-group`. Use o parâmetro `parameter-name-values` para especificar `reserved-memory` e um valor para ele.

O seguinte exemplo de CLI modifica o `parameter group` personalizado `redis32-m3x1` para que ele use `reserved-memory` para gerenciar a memória reservada. Um valor deve ser atribuído a `ParameterValue` para que o grupo de parâmetros use o parâmetro `ParameterName` para gerenciamento de memória reservada. Como a versão do mecanismo é mais recente do que 2.8.22, definimos o valor como `3565158400`, que equivale a 25% da `maxmemory` de um `cache.m3.xlarge`. Para obter mais informações, consulte [Modificar um parameter group](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

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 causa disso, recomendamos manter a utilização da CPU abaixo de 80 por cento em instâncias de vários núcleos e abaixo de 50 por cento em instâncias de núcleo único durante a refragmentação. Monitore as métricas do ElastiCache para Redis e inicie a refragmentação antes que a sua aplicação comece a observar problemas de escalabilidade. As métricas úteis para acompanhar são `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` e `BytesUsedForCacheItems`.
- **Garanta memória livre suficiente disponível antes da redução de escala na horizontal:** se você estiver reduzindo a escala na horizontal, garanta que a memória livre disponível nos fragmentos a serem retidos é, pelo menos, 1,5 vez maior do que a memória usada nos fragmentos que você planeja remover.
- **Inicie a refragmentação em horários fora de pico:** essa prática ajuda a reduzir a latência e o impacto de throughput no cliente durante a operação de refragmentação. Ela também ajuda a concluir a refragmentação com mais rapidez à medida que mais recursos podem ser usados na redistribuição de slots.
- **Analise o comportamento de tempo limite do cliente:** alguns clientes podem observar maior latência durante o redimensionamento de cluster online. Configurar sua biblioteca de cliente com um tempo limite maior pode ajudar dando tempo para o sistema se conectar mesmo em condições de carga maiores no servidor. Em alguns casos, você pode abrir um grande número de conexões com o servidor. Nesses casos, considere adicionar o recuo exponencial para uma nova conexão lógica. Fazer isso pode ajudar a evitar uma intermitência de novas conexões acessando o servidor ao mesmo tempo.
- **Carregue suas funções em cada fragmento:** ao aumentar a escala horizontalmente do cluster, o ElastiCache replicará automaticamente as funções carregadas em um dos nós existentes (selecionados aleatoriamente) para os novos nós. Se seu cluster tiver o Redis 7.0 ou posterior e sua aplicação usar [Redis Functions](#) (Funções do Redis), recomendamos carregar todas as suas funções em todos os fragmentos antes de aumentar a escala horizontalmente para que seu cluster não tenha funções diferentes em fragmentos diferentes.

Após a refragmentação, observe o seguinte:

- A redução da escala horizontalmente pode ser parcialmente bem-sucedida se não houver memória suficiente disponível nos fragmentos de destino. Se isso ocorrer, analise a memória disponível e refaça a operação, se necessário. Os dados nos fragmentos de destino não serão excluídos.
- 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 reestilhaçamento. Antes do Redis 6, o comando BRPOPLPUSH não é compatível caso opere no slot que está sendo migrado.

Minimização do tempo de inatividade durante a manutenção

A configuração do modo de cluster tem a melhor disponibilidade durante operações gerenciadas e não gerenciadas. Recomendamos usar um cliente compatível com o modo de cluster que se conecta ao endpoint de descoberta de cluster. Para o modo de cluster desabilitado recomendamos usar o endpoint principal para todas as operações de gravação.

Para atividades de leitura, os aplicativos também podem se conectar a qualquer nó no cluster. Ao contrário do endpoint primário, endpoints de nó são resolvidos para endpoints específicos. Se você fizer uma alteração no cluster, como adicionar ou excluir uma réplica, deverá atualizar os endpoints de nó no seu aplicativo. É por isso que, para o modo de cluster desabilitado, recomendamos que você use o endpoint do leitor para a atividade de leitura.

Se o failover automático estiver habilitado no cluster, o nó primário poderá ser alterado. Portanto, o aplicativo deve confirmar a função do nó e atualizar todos os endpoints de leitura. Fazer isso ajuda a garantir que não haja sobrecarga no nó principal. Com o failover automático desabilitado, a função do nó não é alterada. No entanto, o tempo de inatividade em operações gerenciadas ou não gerenciadas é maior em comparação ao dos clusters com failover automático habilitado.

Evite direcionar solicitações de leitura para um único nó de réplica de leitura, pois a indisponibilidade pode causar uma interrupção na leitura. Faça o fallback para a leitura pelo primário ou garanta que você tenha pelo menos duas réplicas de leitura para evitar qualquer interrupção de leitura durante a manutenção.

Práticas recomendadas do Redis

Estas são as práticas recomendadas ao usar o Redis para melhorar o desempenho e aumentar a confiabilidade:

- Usar configurações habilitadas para o modo cluster: o modo cluster habilitado permite a escala horizontal do cache para conseguir mais armazenamento e throughput do que uma configuração de modo cluster desabilitado. A tecnologia sem servidor do ElastiCache só permanece disponível em uma configuração de modo cluster habilitado.
- Usar conexões de longa duração: a criação de uma nova conexão é cara e consome tempo e recursos de CPU do cache. Reutilize conexões quando possível (por exemplo, com agrupamento de conexões) para amortizar esse custo em muitos comandos.
- Ler a partir de réplicas: se você estiver usando a tecnologia sem servidor do ElastiCache ou tiver réplicas de leitura provisionadas (clusters autoprojeto), direcione leituras para réplicas a fim de obter uma escalabilidade melhor e/ou menos 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.

No ElastiCache sem servidor, a leitura pela porta de réplica (6380) vai direcionar as leituras para a zona de disponibilidade local do cliente quando possível, o que reduz 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.
- Usar pub/sub fragmentado: ao usar o Redis para dar suporte a workloads pub/sub com throughput elevado, é recomendável usar [pub/sub fragmentado](#) (disponível com o Redis 7 ou posterior). O pub/sub tradicional em clusters habilitados para modo cluster transmite mensagens para todos

os nós do cluster, o que pode acarretar `EngineCPUUtilization` elevada. No ElastiCache sem servidor, os comandos `pub/sub` tradicionais usam internamente comandos `pub/sub` fragmentados.

Estratégias de armazenamento em cache

No tópico a seguir, você pode encontrar estratégias para preencher e manter seu cache.

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](#)
- [Adicionar o 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.

O Amazon ElastiCache é um repositório de chaves-valores na memória entre a aplicação e o armazenamento de dados (banco de dados) que ele acessa. Sempre que a sua aplicação solicita dados, ela primeiro faz a solicitação ao cache do ElastiCache. Se os dados existirem no cache e forem atuais, o ElastiCache os retornará à sua aplicação. Se os dados não existirem no cache ou tiverem expirado, sua aplicação solicitará os dados do seu armazenamento de dados. Seu armazenamento de dados retorna os dados para sua aplicação. Sua aplicação, em seguida, grava os dados recebidos do armazenamento no cache. Dessa forma, eles podem ser recuperados mais rapidamente na próxima vez que forem solicitados.

Um acerto de cache ocorre quando os dados estão no cache e não estão expirados:

1. A aplicação solicita dados do cache.
2. O cache retorna os dados à aplicação.

Uma perda de cache ocorre quando os dados não estão no cache ou estão expirados:

1. Sua aplicação solicita dados do cache.

2. O cache não possui os dados solicitados e, portanto, retorna `null`.
3. Sua aplicação solicita e recebe os dados do banco de dados.
4. A aplicação atualiza o cache com os novos dados.

Vantagens e desvantagens do carregamento lento

As vantagens do carregamento lento são as seguintes:

- Apenas dados solicitados são armazenados em cache.

Como a maioria dos dados nunca é solicitada, o carregamento lento evita o preenchimento do cache com dados que não são solicitados.

- Falhas de nó não são fatais para sua aplicação.

Quando um nó falha e é substituído por um novo nó vazio, sua aplicação continua a funcionar, embora com latência aumentada. À medida que as solicitações são feitas para o novo nó, cada perda de cache resulta em uma consulta do banco de dados. Ao mesmo tempo, a cópia de dados é adicionada ao cache para que as solicitações subsequentes sejam recuperadas do cache.

As desvantagens do carregamento lento são as seguintes:

- Existe uma penalidade de ausência no cache. A perda de cache resulta em 3 viagens,
 1. Solicitação inicial de dados do cache
 2. Consulta do banco de dados para os dados
 3. Gravação de dados no cache

Essas perdas podem causar um atraso observável na obtenção dos dados para a aplicação.

- Dados obsoletos.

Se os dados forem gravados no cache somente quando houver uma perda de cache, os dados no cache poderão ficar obsoletos. Esse resultado ocorre porque não há atualizações para o cache quando os dados são alterados no banco de dados. Para resolver esse problema, você pode usar as estratégias [Gravação simultânea](#) e [Adicionar o TTL](#).

Exemplo de pseudocódigo de carregamento lento

O código a seguir é um exemplo de pseudocódigo de lógica de carregamento lento.

```
// *****  
// function that returns a customer's record.  
// Attempts to retrieve the record from the cache.  
// If it is retrieved, the record is returned to the application.  
// If the record is not retrieved from the cache, it is  
//   retrieved from the database,  
//   added to the cache, and  
//   returned to the application  
// *****  
get_customer(customer_id)  
  
    customer_record = cache.get(customer_id)  
    if (customer_record == null)  
  
        customer_record = db.query("SELECT * FROM Customers WHERE id = {0}",  
customer_id)  
        cache.set(customer_id, customer_record)  
  
    return customer_record
```

Para este exemplo, o código da aplicação que obtém os dados é o seguinte.

```
customer_record = get_customer(12345)
```

Gravação simultânea

A estratégia de gravação simultânea adiciona dados ou atualiza dados no cache sempre que eles são gravados no banco de dados.

Vantagens e desvantagens da gravação simultânea

As vantagens da gravação simultânea são as seguintes:

- Os dados no cache nunca são obsoletos.

Como os dados no cache são atualizados sempre que são escritos no banco de dados, eles sempre estão atualizados.

- Penalidade de gravação versus penalidade de leitura.

Cada gravação envolve duas viagens:

1. Uma gravação no cache

2. Uma gravação no banco de dados

Que acrescenta latência ao processo. Dito isto, os usuários finais geralmente são mais tolerantes à latência ao atualizarem dados do que ao 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 tempo de vida \(TTL\)](#), você pode minimizar o espaço desperdiçado.

Exemplo de pseudocódigo de gravação simultânea

O código a seguir é um exemplo de pseudocódigo da lógica de gravação simultânea.

```
// *****  
// function that saves a customer's record.  
// *****  
save_customer(customer_id, values)  
  
    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)  
    cache.set(customer_id, customer_record)  
    return success
```

Para este exemplo, o código da aplicação que obtém os dados é o seguinte.

```
save_customer(12345, {"address": "123 Main"})
```

Adicionar o TTL

O carregamento lento permite dados obsoletos, mas não falha com nós vazios. A gravação simultânea garante que os dados sempre sejam atuais, mas pode falhar com nós vazios e pode preencher o cache com dados supérfluos. Ao adicionar um valor de tempo de vida (TTL) a cada gravação, você pode ter as vantagens de cada estratégia. Ao mesmo tempo, você pode em grande parte evitar desordenar o cache com dados adicionais.

O Time to live (TTL) (Tempo de vida) é um valor inteiro que especifica o número de segundos até a chave expirar. O Redis pode especificar segundos ou milissegundos para este valor. 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 o [set comando](#).

Exemplos de pseudocódigo de TTL

O código a seguir é um exemplo de pseudocódigo da lógica de gravação simultânea com TTL.

```
// *****  
// function that saves a customer's record.  
// The TTL value of 300 means that the record expires  
//   300 seconds (5 minutes) after the set command  
//   and future reads will have to query the database.  
// *****  
save_customer(customer_id, values)  
  
    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)  
    cache.set(customer_id, customer_record, 300)  
  
    return success
```

O código a seguir é um exemplo de pseudocódigo de lógica de carregamento lento com TTL.

```
// *****  
// function that returns a customer's record.  
// Attempts to retrieve the record from the cache.
```

```
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//   retrieved from the database,
//   added to the cache, and
//   returned to the application.
// The TTL value of 300 means that the record expires
//   300 seconds (5 minutes) after the set command
//   and subsequent reads will have to query the database.
// *****
get_customer(customer_id)

    customer_record = cache.get(customer_id)

    if (customer_record != null)
        if (customer_record.TTL < 300)
            return customer_record          // return the record and exit function

    // do this only if the record did not exist in the cache OR
    //   the TTL was >= 300, i.e., the record in the cache had expired.
    customer_record = db.query("SELECT * FROM Customers WHERE id = {0}", customer_id)
    cache.set(customer_id, customer_record, 300) // update the cache
    return customer_record          // return the newly retrieved record and exit
function
```

Para este exemplo, o código da aplicação que obtém os dados é o seguinte.

```
save_customer(12345, {"address": "123 Main"})
```

```
customer_record = get_customer(12345)
```

Tópicos relacionados da

- [Armazenamento de dados na memória](#)
- [Escolha de um mecanismo e versão](#)
- [Dimensionamento ElastiCache para Redis](#)

Gerenciamento do cluster autoprojetoado

Esta seção contém tópicos que ajudam você a gerenciar os clusters autoprojetoados.

Note

Esses tópicos não se aplicam ao ElastiCache sem servidor.

Tópicos

- [Auto Scaling ElastiCache para clusters Redis](#)
- [Modificar o modo de cluster](#)
- [Replicação em regiões da AWS usando o armazenamentos de dados globais](#)
- [Alta disponibilidade com o uso de grupos de replicação](#)
- [Gerenciamento da manutenção](#)
- [Configuração de parâmetros do mecanismo usando grupos de parâmetros](#)

Auto Scaling ElastiCache para clusters Redis

Pré-requisitos

ElastiCache para Redis, o Auto Scaling está limitado ao seguinte:

- Clusters do Redis (modo cluster habilitado) que executem o mecanismo Redis versão 6.0 em diante
- Clusters em camadas de dados (modo cluster ativado) que executem o mecanismo Redis versão 7.0.7 em diante
- Tamanhos de instância: Large, XLarge, 2XLarge
- Famílias de tipo de instância: R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- O Auto Scaling in ElastiCache for Redis não é compatível com clusters executados em datastores globais, Outposts ou Locais Zones.

Gerenciando a capacidade automaticamente com o ElastiCache Redis Auto Scaling

ElastiCache para Redis, o auto scaling é a capacidade de aumentar ou diminuir automaticamente os fragmentos ou réplicas desejados em seu serviço ElastiCache for Redis. ElastiCache for Redis utiliza 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 for Redis, o auto scaling usa a política para aumentar ou diminuir o número de instâncias em resposta às cargas de trabalho reais.

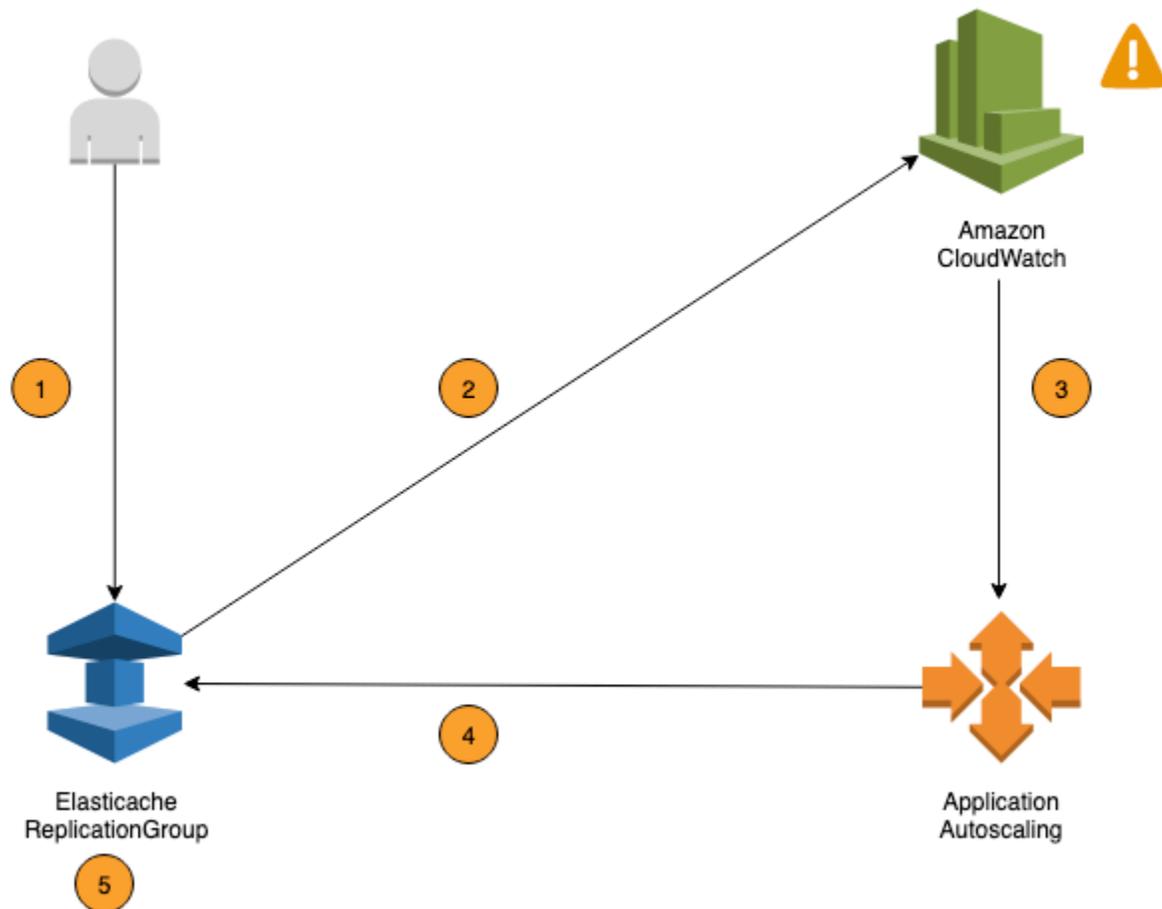
Você pode usar o AWS Management Console para aplicar uma política de escalabilidade com base em uma métrica predefinida. Uma `predefined metric` é definida em uma enumeração, para que você possa especificá-la por nome no código ou usá-la no AWS Management Console. As métricas personalizadas não estão disponíveis para seleção ao usar o AWS Management Console. Como alternativa, você pode usar a API Application Auto Scaling AWS CLI ou a Application Auto Scaling para aplicar uma política de escalabilidade com base em uma métrica predefinida ou personalizada.

ElastiCache for Redis suporta escalabilidade para as seguintes dimensões:

- **Fragmentos:** adicionar/remover fragmentos automaticamente no cluster semelhantes à refragmentação manual online. Nesse caso, ElastiCache para o Redis, o escalonamento automático aciona o escalonamento em seu nome.
- **Replicas (Réplicas):** adicionar/remover automaticamente réplicas no cluster semelhantes às operações manuais de aumentar/diminuir réplicas. ElastiCache para Redis, o auto scaling adiciona/remove réplicas uniformemente em todos os fragmentos do cluster.

ElastiCache for Redis oferece suporte aos 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 escalabilidade automática do Application ElastiCache for 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 ElastiCache de escalonamento automático do Redis, conforme mostrado no diagrama anterior:

1. Você cria uma política de auto scaling ElastiCache for Redis para seu grupo de replicação ElastiCache for Redis.
2. ElastiCache for Redis, o auto scaling cria um par de CloudWatch alarmes em seu nome. Cada par representa seus limites superiores e inferiores para métricas. Esses CloudWatch alarmes são acionados quando a utilização real do cluster se desvia da utilização desejada por um período prolongado. 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 do Redis ElastiCache para avaliar sua política de escalabilidade.
4. ElastiCache for Redis, o auto scaling emite uma solicitação de modificação para ajustar a capacidade do cluster.

5. ElastiCache for Redis 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 Redis Auto Scaling funciona, suponha que você tenha um cluster chamado `UsersCluster`. Ao monitorar as CloudWatch métricas `UsersCluster`, você determina o máximo de fragmentos que o cluster exige quando o tráfego está no pico e o mínimo de fragmentos quando o tráfego está no ponto mais baixo. Você também decide um valor de destino para utilização da CPU para o cluster `UsersCluster`. ElastiCache for Redis, 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 igual ou próxima ao valor alvo.

Note

O escalonamento pode levar um tempo perceptível e exigirá recursos extras do cluster para que os fragmentos se rebalanceiem. ElastiCache para Redis Auto Scaling modifica as configurações de recursos somente quando a carga de trabalho real permanece elevada (ou deprimida) por um período sustentado de vários minutos. O algoritmo ElastiCache de rastreamento de metas de escalonamento automático do Redis busca manter a utilização desejada igual ou próxima ao valor escolhido a longo prazo.

Políticas de Auto Scaling

Uma política de escalabilidade tem os seguintes componentes:

- Uma métrica de destino: a métrica do CloudWatch que a escalabilidade automática do ElastiCache for Redis usa para determinar o momento e a quantidade a escalar.
- Capacidade mínima e máxima: o número mínimo e máximo de fragmentos a ser usado para a escalabilidade.

Important

Ao criar a política de Auto Scaling, se a capacidade atual for maior do que a capacidade máxima configurada, reduzimos a escala na horizontal até `MaxCapacity` durante a criação da política. Da mesma forma, se a capacidade atual for menor do que a capacidade mínima configurada, aumentamos a escala na horizontal até `MinCapacity`.

- Um período de desaquecimento: a quantidade de tempo, em segundos, que outra ação de aumento ou redução de escala na horizontal deve aguardar antes de ser iniciada, após a conclusão de uma ação de expansão ou redução.
- Uma função vinculada a serviços: uma função do Identity and Access Management (IAM) da AWS vinculada a um serviço AWS específico. Uma função vinculada ao serviço inclui todas as permissões que o serviço exige para chamar todos os outros serviços da AWS em seu nome. O Auto Scaling do ElastiCache for Redis gera automaticamente essa função, `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`, para você.
- Habilitar ou desabilitar atividades de redução de escala na horizontal: capacidade de habilitar ou desabilitar atividades de redução de escala na horizontal para uma política.

Tópicos

- [Métrica de destino para o Auto Scaling](#)
- [Capacidades mínima e máxima](#)
- [Período de desaquecimento](#)
- [Habilitar ou desabilitar atividades de redução](#)

Métrica de destino para o Auto Scaling

Neste tipo de política, uma métrica predefinida ou personalizada e um valor de destino dessa métrica são especificados na configuração de uma política de escalabilidade de rastreamento de destino. O Auto Scaling do ElastiCache for Redis cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove fragmentos/réplicas conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a workloads variáveis. Essas políticas também minimizam flutuações rápidas no número de fragmentos/réplicas disponíveis para seu cluster.

Por exemplo, considere uma política de escalabilidade que use a métrica predefinida de média `ElastiCachePrimaryEngineCPUUtilization`. Essa política pode manter a utilização da CPU a um percentual de utilização específico, por exemplo, 70 por cento, ou próximo a isso.

Note

Para cada cluster, você pode criar somente uma política de Auto Scaling para cada métrica de destino.

Capacidades mínima e máxima

Estilhaços

Você pode especificar o número máximo de fragmentos que podem ser escalados pelo Auto Scaling do ElastiCache for 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 Auto Scaling do ElastiCache for Redis. O valor mínimo é 1, e deve ser menor ou igual ao valor especificado para o número máximo de fragmentos de 250.

Réplicas

Você pode especificar o número máximo de réplicas a serem gerenciadas pelo Auto Scaling do ElastiCache for Redis. Esse valor deve ser igual ou menor que 5. Você também pode especificar o número mínimo de réplicas a serem gerenciadas pelo Auto Scaling do ElastiCache for Redis. O valor mínimo é 1, e deve ser menor ou igual ao valor especificado para o número máximo de réplicas de 5.

Para determinar o número mínimo e máximo de fragmentos/réplicas que você precisa para um tráfego típico, teste a configuração do Auto Scaling com a taxa esperada de tráfego para o modelo.

Note

As políticas de escalabilidade automática do ElastiCache for Redis aumentam a capacidade do cluster até atingir o tamanho máximo definido ou até que os limites do serviço sejam aplicados. Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite Nodes per cluster per instance type (Nós por cluster por tipo de instância).

⚠ Important

Redução de escala na horizontal quando não há tráfego Se o tráfego de uma variante atingir zero, o ElastiCache for Redis automaticamente reduz a escala na horizontal para o número mínimo de instâncias especificado.

Período de desaquecimento

Você pode ajustar a responsividade das políticas de escalabilidade com monitoramento do objetivo, adicionando períodos de desaquecimento que afetam a escalabilidade de seu cluster. Um desaquecimento bloqueia solicitações subsequentes de redução ou expansão até o período expirar. Isso atrasa as exclusões de fragmentos/réplicas no cluster do ElastiCache for Redis para solicitações de redução de escala na horizontal e a criação de fragmentos/réplicas para solicitações de aumento de escala na horizontal. Você pode especificar os seguintes desaquecimentos:

- Uma atividade de redução de escala na horizontal diminui o número de fragmentos/réplicas no seu cluster do ElastiCache for Redis. 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 aumento de escala na horizontal aumenta o número de fragmentos/réplicas no seu cluster do ElastiCache for Redis. Um desaquecimento de expansão especifica a quantidade de tempo, em segundos, após a conclusão de uma ação de expansão antes que quaisquer outras atividades de expansão possam iniciar.

Quando o período de desaquecimento de redução ou aumento de escala na horizontal não é especificado, o padrão para o aumento é de 600 segundos, e para a redução, 900 segundos.

Habilitar ou desabilitar atividades de redução

Você pode habilitar ou desabilitar atividades de redução para as políticas. Habilitar atividades de redução de escala na horizontal permite que as políticas de escalabilidade excluam fragmentos/réplicas. Quando as atividades de redução são habilitadas, o desaquecimento de redução na política de escalabilidade aplica-se a atividades de redução. Desabilitar atividades de redução de escala na horizontal evita que as políticas de escalabilidade excluam fragmentos/réplicas.

Note

As atividades de aumento de escala na horizontal são habilitadas para que a política de escalabilidade possa criar fragmentos/réplicas do ElastiCache for Redis, conforme necessário.

Permissões do IAM necessárias ElastiCache para o Redis Auto Scaling

ElastiCache para Redis, o Auto Scaling é possível graças a uma combinação das ElastiCache APIs for Redis CloudWatch, e Application Auto Scaling. Os clusters são criados e atualizados com o ElastiCache for Redis, os alarmes são criados com e as políticas de escalabilidade são criadas com CloudWatch o Application Auto Scaling. Além das permissões padrão do IAM para criar e atualizar clusters, o usuário do IAM que acessa ElastiCache as configurações do Redis Auto Scaling deve ter as permissões apropriadas para os serviços que oferecem suporte à escalabilidade dinâmica. Os usuários do IAM precisam ter as permissões para usar as ações exibidas na política de exemplo 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 auto scaling ElastiCache for Redis também precisa de permissão para descrever seus clusters e CloudWatch alarmes, além de permissões para modificar sua ElastiCache capacidade alvo do Redis em seu nome. Se você habilitar o Auto Scaling ElastiCache para seu cluster for Redis, ele criará uma função vinculada ao serviço chamada `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`. Essa função vinculada ao serviço ElastiCache concede ao Redis permissão de escalonamento 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 ElastiCache para o escalonamento automático do Redis. Para obter mais informações, consulte [Funções vinculadas a serviços ElastiCache para escalonamento automático do Redis no Guia do usuário do Application Auto Scaling](#).

Práticas recomendadas de escalabilidade automática

Antes de se registrar no Auto Scaling, recomendamos o seguinte:

1. Use apenas uma métrica de monitoramento: identifique se o cluster tem workloads com uso intenso da CPU ou de dados e use uma métrica predefinida correspondente para definir a política de ajuste de escala.
 - CPU do mecanismo: `ElastiCachePrimaryEngineCPUUtilization` (dimensão fragmentada) ou `ElastiCacheReplicaEngineCPUUtilization` (dimensão da réplica)
 - Uso do banco de dados:
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` essa política de ajuste de escala funciona melhor com `maxmemory-policy` definido como `noeviction` no cluster.

Recomendamos que você evite várias políticas por dimensão no cluster. ElastiCache para Redis, o Auto Scaling expandirá a meta escalável se alguma política de rastreamento de metas estiver pronta para ser expandida, mas será ampliada somente se todas as políticas de rastreamento

de metas (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 o Redis 7.0 ou posterior e sua aplicação usar [Redis Functions](#) (Funções do Redis), recomendamos carregar todas as suas funções em todos os fragmentos antes de aumentar a escala horizontalmente para que seu cluster não tenha funções diferentes em fragmentos diferentes.

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

O tópico seguinte fornece detalhes sobre o monitoramento do objetivo e as políticas programadas e como aplicá-las usando o AWS Management Console, a AWS CLI e APIs.

Políticas de escalabilidade de rastreamento de destino

Com as políticas de dimensionamento com monitoramento do objetivo, você seleciona uma métrica e define um valor pretendido. O Auto Scaling do ElastiCache for Redis cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove fragmentos conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a um padrão de carga de flutuação, e minimiza as flutuações rápidas na capacidade da frota.

Por exemplo, considere uma política de escalabilidade que use a métrica predefinida de média `ElastiCachePrimaryEngineCPUUtilization` com valor de destino configurado. Essa política pode manter a utilização da CPU em, ou próxima do valor de destino especificado.

Métricas predefinidas

Uma métrica predefinida é uma estrutura que se refere a um nome, dimensão e estatística (average) específicos de uma determinada métrica do CloudWatch. Sua política de ajuste de escala automático define as seguintes métricas predefinidas para seu cluster:

Nome da métrica predefinida	Nome da métrica do CloudWatch	Dimensão da métrica do CloudWatch.	Tipos de instância inelegíveis
<code>ElastiCachePrimaryEngineCPUUtilization</code>	<code>EngineCPUUtilization</code>	ReplicationGroupId, Função = primária	Nenhum
<code>ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage</code>	<code>DatabaseCapacityUsageCountedForEvictPercentage</code>	Métricas do grupo de replicação do Redis	Nenhum

Nome da métrica predefinida	Nome da métrica do CloudWatch	Diimensão da métrica do CloudWatch.	Tipos de instância inelegíveis
ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage	DatabaseMemoryUsageCountedForEvictPercentage	Métricas do grupo de replicação do Redis	R6gd

Os tipos de instância em camadas de dados não podem usar `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, pois esses tipos de instância armazenam dados na memória e no SSD. O caso de uso previsto para instâncias em camadas de dados é usar 100% de memória e preencher o SSD conforme necessário.

Critérios do Auto Scaling para fragmentos

Quando o serviço detectar que sua métrica predefinida é igual ou maior que a configuração de destino, ele aumentará automaticamente a capacidade dos fragmentos. O ElastiCache for Redis aumenta a escala na horizontal dos fragmentos do seu cluster em uma contagem igual à maior de dois números: variação percentual do objetivo e 20% dos fragmentos atuais. Para reduzir a escala na horizontal, o ElastiCache for Redis não fara a redução automaticamente a menos que o valor geral da métrica seja inferior a 75% do seu objetivo definido.

Para um exemplo de aumento de escala na horizontal, se você tiver 50 fragmentos e

- se o seu objetivo violar em 30%, o ElastiCache para Redis aumentará a escala na horizontal em 30%, o que resultará em 65 fragmentos por cluster.
- se o seu objetivo violar em 10%, o ElastiCache para Redis aumentará a escala na horizontal em um mínimo padrão de 20%, o que resultará em 60 fragmentos por cluster.

Para um exemplo de redução de escala na horizontal, se você selecionou um valor de objetivo de 60%, o ElastiCache for Redis não reduzirá a escala na horizontal automaticamente até que a métrica seja menor ou igual a 45% (25% abaixo do objetivo de 60%).

Considerações sobre o Auto Scaling

Lembre-se das seguintes considerações:

- Uma política de escalabilidade de rastreamento de destino pressupõe que ela deve aumentar a escalabilidade quando a métrica especificada estiver acima do valor de destino. Você não pode usar uma política de escalabilidade de rastreamento de destino para expandir quando a métrica especificada estiver abaixo do valor de destino. O ElastiCache for Redis aumenta a escala na horizontal de fragmentos em um desvio mínimo de 20% do objetivo dos fragmentos existentes no cluster.
- Uma política de escalabilidade de rastreamento de destino não escala quando a métrica especificada tem dados insuficientes. Ela não reduz a escala horizontalmente, porque não interpreta dados insuficientes como baixa utilização.
- Você pode ver lacunas entre o valor de destino e os pontos de dados de métrica reais. Isso ocorre porque o Auto Scaling do ElastiCache for Redis funciona de maneira segura por arredondamento para cima ou para baixo, quando ele determina a capacidade a ser adicionada ou removida. Isso evita que ele adicione capacidade insuficiente ou remova muita capacidade.
- Para garantir a disponibilidade da aplicação, o serviço aumenta a escala na horizontal proporcionalmente à métrica o mais rápido possível, mas é reduz a escala na horizontal de forma mais conservadora.
- Você pode ter várias políticas de escalabilidade de monitoramento do objetivo para um cluster do ElastiCache for Redis, desde que cada uma delas use uma métrica diferente. A intenção do Auto Scaling do ElastiCache for Redis é sempre priorizar a disponibilidade, portanto, seu comportamento será diferente dependendo se as políticas de monitoramento do objetivo estão prontas para aumentar ou reduzir a escala na horizontal. Ele vai aumentar o serviço se qualquer uma das políticas de monitoramento do objetivo estiverem prontas para aumentar, mas vai reduzir somente se todas as políticas de monitoramento do objetivo (com a parte de redução habilitada) estiverem prontas para reduzir.
- Não edite nem exclua os alarmes do CloudWatch que o Auto Scaling do serviço gerencia para uma política de escalabilidade com monitoramento do objetivo. O Auto Scaling do ElastiCache for Redis exclui os alarmes automaticamente quando você exclui a política de escalabilidade.
- O Auto Scaling do ElastiCache for Redis não impede a modificação manual de fragmentos do cluster. Esses ajustes manuais não afetam alarmes existentes do CloudWatch que estejam conectados à política de escalabilidade, mas podem afetar métricas que possam acionar esses alarmes do CloudWatch.

- Esses alarmes do CloudWatch gerenciados pelo Auto Scaling são definidos na métrica do AVG em todos os fragmentos do cluster. Assim, ter fragmentos quentes pode resultar em qualquer cenário de:
 - escalabilidade quando não for necessário devido à carga em alguns fragmentos quentes que acionam um alarme do CloudWatch
 - não escalar quando necessário devido ao AVG agregado em todos os fragmentos que afetam o alarme não violarem.
- Os limites padrão do ElastiCache for Redis em nós por cluster ainda se aplicam. Então, ao optar pelo Auto Scaling, se você espera que os nós máximos sejam mais do que o limite padrão, solicite um aumento de limite em [Limites de serviço da AWS](#) e escolha o tipo de limite Nós por cluster por tipo de instância.
- Certifique-se de que você tenha ENIs (interfaces de rede elásticas) suficientes disponíveis em sua VPC, que são necessárias durante o aumento de escala na horizontal. Para obter mais informações, consulte [Interfaces de rede elástica](#).
- Se não houver capacidade suficiente disponível do EC2, a autoescalabilidade do ElastiCache para Redis não aumentará a escala e atrasará até que a capacidade esteja disponível.
- O Auto Scaling do ElastiCache for Redis, durante a redução de escala na horizontal, não removerá fragmentos com slots com um tamanho de item maior que 256 MB pós-serialização.
- Durante a redução de escala na horizontal, ele não removerá fragmentos se houver memória insuficiente disponível na configuração de fragmento resultante.

Adicionar uma política de escalabilidade

Você pode adicionar uma política de escalabilidade usando o AWS Management Console.

Para adicionar uma política do Auto Scaling a um cluster do ElastiCache for Redis

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
3. Selecione o cluster ao qual você deseja adicionar uma política (escolha o botão à esquerda do nome do cluster).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Escolha add dynamic scaling (adicionar escalabilidade dinâmica).
6. Em Policy Name (Nome da política), insira um nome para a política.

7. Em Scalable Dimension (Dimensão escalável), escolha shards (fragmentos).
8. Quanto à métrica de destino, escolha uma das seguintes opções:
 - Primary CPU Utilization (Utilização da CPU primária) para criar uma política com base na utilização média da CPU.
 - Memory (Memória) para criar uma política com base na memória média do banco de dados.
 - Capacidade de criar uma política com base no uso médio da capacidade do banco de dados. A métrica de capacidade inclui a utilização de memória e SSD para instâncias em camadas de dados e a utilização da memória para todos os outros tipos de instância.
9. Para o valor alvo, escolha um valor maior ou igual a 35 e menor ou igual a 70. O ajuste de escala automático manterá esse valor para a métrica de destino selecionada em seus fragmentos do ElastiCache:
 - Utilização primária da CPU: mantém o valor alvo da métrica `EngineCPUUtilization` nos nós primários.
 - Memória: mantém o valor alvo da métrica `DatabaseMemoryUsageCountedForEvictPercentage`
 - Capacidade: mantém o valor alvo da métrica `DatabaseCapacityUsageCountedForEvictPercentage`

Os fragmentos do cluster serão adicionados ou removidos para manter a métrica próxima ao valor especificado.
10. (Opcional) Não há suporte do console para os períodos de desaquecimento de redução ou aumento de escala na horizontal. Use a AWS CLI para modificar os valores de desaquecimento.
11. Em Minimum capacity (Capacidade mínima), insira o número mínimo de fragmentos que o Auto Scaling do ElastiCache for Redis é solicitado a manter.
12. Em Maximum capacity (Capacidade máxima), insira o número máximo de fragmentos que o Auto Scaling do ElastiCache for Redis é solicitado a manter. Esse valor deve ser igual ou menor que 250.
13. Escolha Create (Criar).

Registro de um destino escalável

Antes de usar o Auto Scaling do com um cluster do ElastiCache for Redis, você deve registrar seu cluster na escalabilidade automática do ElastiCache for Redis. Faça isso para definir a dimensão de

escalabilidade e os limites a serem aplicados no cluster. A escalabilidade automática do ElastiCache for Redis escala dinamicamente o cluster do ElastiCache for Redis ao longo da dimensão escalável `elasticache:replication-group:NodeGroups`, que representa o número de fragmentos de cluster.

Como usar a AWS CLI

Para registrar o cluster do ElastiCache for Redis, use o comando [register-scalable-target](#) com os seguintes parâmetros:

- `--service-namespace`: defina o valor como `elasticache`
- `--resource-id`: o identificador do recurso para o cluster do ElastiCache for Redis. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache for Redis, 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 Auto Scaling do ElastiCache for Redis. 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 Auto Scaling do ElastiCache for Redis. Para obter informações sobre a relação entre `--min-capacity`, `--max-capacity` e o número de fragmentos em seu cluster, consulte [Capacidades mínima e máxima](#).

Example

No exemplo a seguir, registre um cluster do ElastiCache for Redis de nome `myscalablecluster`. O registro indica que o cluster deve ser escalado dinamicamente para ter de um a dez fragmentos.

Para Linux, macOS ou Unix:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace elasticache \  
  --resource-id replication-group/myscalablecluster \  
  --scalable-dimension elasticache:replication-group:NodeGroups \  
  --min-capacity 1 \  
  --max-capacity 10 \  

```

Para Windows:

```
aws application-autoscaling register-scalable-target ^
  --service-namespace elasticache ^
  --resource-id replication-group/myscalablecluster ^
  --scalable-dimension elasticache:replication-group:NodeGroups ^
  --min-capacity 1 ^
  --max-capacity 10 ^
```

Uso da API

Para registrar o cluster do ElastiCache, use o comando [register-scalable-target](#) com os seguintes parâmetros:

- **ServiceNamespace:** defina este valor como `elasticache`.
- **ResourceID:** o identificador de recursos para o cluster do ElastiCache. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache for Redis, por exemplo `replication-group/myscalablecluster`.
- **ScalableDimension:** defina este valor como `elasticache:replication-group:NodeGroups`.
- **MinCapacity:** o número mínimo de fragmentos a serem gerenciados pelo Auto Scaling do ElastiCache for 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 fragmentos a serem gerenciados pelo Auto Scaling do ElastiCache for 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, registre um cluster do ElastiCache for Redis chamado `myscalablecluster` na API do Application Auto Scaling. Este registro indica que o cluster deve ser escalado dinamicamente para ter de uma a 5 réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
```

```
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definir uma política de ajuste de escala

Uma configuração de política de escalabilidade de rastreamento de destino é representada por um bloco JSON no qual as métricas e valores de destino são definidos. Você pode salvar uma configuração de política de escalabilidade como um bloco JSON em um arquivo de texto. Você usa esse arquivo de texto ao invocar a AWS CLI ou a API Application Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de políticas, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API Application Auto Scaling.

As seguintes opções estão disponíveis para definir uma configuração de política de escalabilidade com monitoramento do objetivo:

Tópicos

- [Uso de uma métrica predefinida](#)
- [Uso de uma métrica personalizada](#)
- [Uso de períodos de desaquecimento](#)
- [Desabilitar a atividade de redução](#)
- [Aplicar uma política de escalabilidade](#)

Uso de uma métrica predefinida

Usando métricas predefinidas, você pode definir rapidamente uma política de escalabilidade de rastreamento de metas ElastiCache para um cluster do Redis que funciona com o rastreamento de metas no Auto Scaling do Redis ElastiCache .

Atualmente, o ElastiCache for Redis oferece suporte às seguintes métricas predefinidas no Redis ElastiCache Auto Scaling NodeGroup :

- **ElastiCachePrimaryEngineCPUUtilization** — O valor médio da **EngineCPUUtilization** métrica em CloudWatch todos os nós primários no cluster ElastiCache for Redis.

- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— O valor médio da `DatabaseMemoryUsageCountedForEvictPercentage` métrica em CloudWatch todos os nós primários no cluster ElastiCache for Redis.
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— O valor médio da `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` métrica em CloudWatch todos os nós primários no cluster ElastiCache for Redis.

Para ter mais informações sobre as métricas

`EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` e `DatabaseCapacityUsageCountedForEvictPercentage`, consulte [do uso com métricas do CloudWatch](#). 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 escalabilidade de rastreamento de destinos ElastiCache para um cluster do Redis. Nessa configuração, a métrica `ElastiCachePrimaryEngineCPUUtilization` predefinida é usada para ajustar o cluster ElastiCache for Redis com base em uma utilização média da CPU 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 for Redis com base em uma utilização média da CPU 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 a Referência [TargetTrackingScalingPolicyConfiguration](#) da API Application Auto Scaling.

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 for Redis com base em uma utilização média da CPU 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 do Redis desativando ElastiCache 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 a Referência [TargetTrackingScalingPolicyConfiguration](#) da API Application Auto Scaling.

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica `ElastiCachePrimaryEngineCPUUtilization` predefinida ajusta um cluster ElastiCache for Redis com base em uma utilização média da CPU 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 no Auto Scaling ElastiCache for Redis e definir uma política de escalabilidade, você aplica a política de escalabilidade ao cluster registrado. Para aplicar uma

política de escalabilidade a um ElastiCache cluster do Redis, você pode usar a API Application Auto Scaling AWS CLI ou a Application Auto Scaling.

Aplicando uma política de escalabilidade usando o AWS CLI

Para aplicar uma política de escalabilidade ao seu cluster ElastiCache for 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 Redis. ElastiCache Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster ElastiCache for Redis, 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 ElastiCache cluster for Redis.

No exemplo a seguir, você aplica uma política de escalabilidade de rastreamento de destino nomeada `myscalablepolicy` ElastiCache para um cluster do Redis chamado de `for Redis auto myscalablecluster scaling`. ElastiCache Para fazer isso, use uma configuração de política salva em um arquivo chamado `config.json`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling put-scaling-policy \  
  --policy-name myscalablepolicy \  
  --policy-type TargetTrackingScaling \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:NodeGroups \  
  --target-tracking-scaling-policy-configuration file://config.json
```

Para Windows:

```
aws application-autoscaling put-scaling-policy ^
```

```
--policy-name myscalablepolicy ^
--policy-type TargetTrackingScaling ^
--resource-id replication-group/myscalablecluster ^
--service-namespace elasticache ^
--scalable-dimension elasticache:replication-group:NodeGroups ^
--target-tracking-scaling-policy-configuration file://config.json
```

Aplicação de uma política de escalabilidade usando a API

Para aplicar uma política de escalabilidade ao seu cluster ElastiCache for 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 o Redis. ElastiCache Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do cluster ElastiCache for Redis, 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 ElastiCache cluster for Redis.

No exemplo a seguir, você aplica uma política de escalabilidade de rastreamento de destino nomeada `myscalablepolicy` ElastiCache para um cluster do Redis chamado de `for Redis auto myscalablecluster scaling`. 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, a AWS CLI ou a API do Application Auto Scaling.

Editar uma política de escalabilidade usando o AWS Management Console

Para editar uma política do Auto Scaling em um cluster do ElastiCache for Redis

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
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 Modify (Modificar).

Edição de uma política de escalabilidade usando a AWS CLI e a API

Você pode usar a AWS CLI ou a API do Application Auto Scaling para editar uma política de escalabilidade da mesma forma que você aplica uma política de escalabilidade:

- Ao usar a AWS CLI, especifique o nome da política que você deseja editar no parâmetro `--policy-name`. Especifique novos valores para os parâmetros que você deseja alterar.

- Ao usar a API do Application Auto Scaling, especifique o nome da política que você deseja editar no parâmetro `PolicyName`. Especifique novos valores para os parâmetros que você deseja alterar.

Para obter mais informações, consulte [Aplicar uma política de escalabilidade](#).

Excluir uma política de escalabilidade

Você pode excluir uma política de escalabilidade usando o AWS Management Console, a AWS CLI ou a API do Application Auto Scaling.

Excluir uma política de escalabilidade usando o AWS Management Console

Para excluir uma política do Auto Scaling de um cluster do ElastiCache for Redis

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
3. Escolha o cluster cuja política de autoescalabilidade você deseja editar (selecione o nome do cluster, e não o botão à esquerda).
4. Selecione a guia Auto Scaling policies (Políticas do Auto Scaling).
5. Na seção Auto scaling policies (Políticas de autoescalabilidade), escolha a respectiva política e selecione Delete (Excluir).

Excluir uma política de escalabilidade usando o AWS CLI

Para excluir uma política de escalabilidade do seu cluster do ElastiCache for Redis, use o comando [delete-scaling-policy](#) AWS CLI com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id`: o identificador do recurso para o cluster do ElastiCache for Redis. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache for Redis, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo `myscalepolicy` é excluída de um cluster do ElastiCache for Redis chamado `myscalecluster`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling delete-scaling-policy \  
  --policy-name myscalablepolicy \  
  --resource-id replication-group/myscalecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:NodeGroups
```

Para Windows:

```
aws application-autoscaling delete-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --resource-id replication-group/myscalecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:NodeGroups
```

Exclusão de uma política de escalabilidade usando a API

Para excluir uma política de escalabilidade do seu cluster do ElastiCache for Redis, use o comando [DeleteScalingPolicy](#) AWS CLI com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--resource-id`: o identificador do recurso para o cluster do ElastiCache for Redis. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache for Redis, por exemplo `replication-group/myscalecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:NodeGroups`.

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo `myscalepolicy` é excluída de um cluster do ElastiCache for Redis chamado `myscalecluster`.

```
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 do Auto Scaling

Este trecho mostra como criar uma política com monitoramento do objetivo e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```

ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
    MinCapacity: 1
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    TargetTrackingScalingPolicyConfiguration:

```

```
PredefinedMetricSpecification:  
  PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization  
  TargetValue: 40
```

Escalabilidade programada

A escalabilidade com base em uma programação permite que você dimensione sua aplicação em resposta a alterações de demanda. Para usar a escalabilidade programada, crie ações programadas, que instruem o ElastiCache for Redis a executar ações de escalabilidade em momentos específicos. Quando cria uma ação programada, você especifica um cluster do ElastiCache for Redis existente, quando a ação de escalabilidade deve ocorrer, a capacidade mínima e a capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

Você só pode criar uma ação programada para clusters do ElastiCache for 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 em uma programação recorrente:

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
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 console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
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 console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
3. Escolha o cluster ao qual você deseja adicionar uma política.
4. Escolha Manage Auto Scaling policies (Gerenciar políticas do Auto Scaling) no menu suspenso Actions (Ações).

5. Escolha a guia Auto Scaling policies (Políticas do Auto Scaling).
6. Na seção Auto scaling policies (Políticas do Auto Scaling), escolha a política do Auto Scaling e depois escolha Delete (Excluir) na caixa de diálogo Actions (Ações).

Para gerenciar a escalabilidade programada usando o AWS CLI

Use as seguintes APIs de application-autoscaling:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Use AWS CloudFormation para criar uma ação programada

Este trecho mostra como criar uma política com monitoramento do objetivo e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade ResourceId com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
    MinCapacity: 1
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
    ScheduledActions:
      - EndTime: '2020-12-31T12:00:00.000Z'
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
          ScheduledActionName: First
          Schedule: 'cron(0 18 * * ? *)'
```

Usar o ajuste de escala automático com réplicas

O tópico seguinte fornece detalhes sobre o monitoramento do objetivo e as políticas programadas e como aplicá-las usando o AWS Management Console, a AWS CLI e APIs.

Políticas de escalabilidade de rastreamento de destino

Com as políticas de dimensionamento com monitoramento do objetivo, você seleciona uma métrica e define um valor pretendido. O Auto Scaling do ElastiCache for Redis cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove réplicas uniformemente ao longo de todos os fragmentos conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a um padrão de carga de flutuação, e minimiza as flutuações rápidas na capacidade da frota.

Critérios do Auto Scaling para réplicas

Sua política do Auto Scaling define a seguinte métrica predefinida para seu cluster:

`ElastiCacheReplicaEngineCPUUtilization`: o limite de utilização de AVG EngineCPU agregado em todas as réplicas usadas pelo ElastiCache for Redis para acionar uma operação de escalabilidade automática. Você pode definir a meta de utilização entre 35% e 70%.

Quando o serviço detectar que sua métrica `ElastiCacheReplicaEngineCPUUtilization` é igual ou maior que a configuração de destino, ele aumentará automaticamente as réplicas nos fragmentos. O ElastiCache for Redis aumenta a escala na horizontal das réplicas do seu cluster em uma contagem igual ao maior de dois números: variação percentual do objetivo e uma réplica. Para reduzir a escala na horizontal, o ElastiCache for Redis não fará a redução automaticamente a menos que o valor geral da métrica seja inferior a 75% do seu objetivo definido.

Para um exemplo de aumento de escala na horizontal, se você tiver 5 fragmentos e 1 réplica cada:

Se o seu objetivo for violado em 30%, o ElastiCache for Redis aumentará a escala na horizontal em 1 réplica (máximo (0,3, padrão 1)) em todos os fragmentos, o que resulta em 5 fragmentos com 2 réplicas cada.

Para um exemplo de redução de escala na horizontal, se você selecionou um valor de objetivo de 60%, o ElastiCache for Redis não reduzirá a escala na horizontal automaticamente até que a métrica seja menor ou igual a 45% (25% abaixo do objetivo de 60%).

Considerações sobre o Auto Scaling

Lembre-se das seguintes considerações:

- Uma política de escalabilidade de rastreamento de destino pressupõe que ela deve aumentar a escalabilidade quando a métrica especificada estiver acima do valor de destino. Você não pode usar uma política de escalabilidade de rastreamento de destino para expandir quando a métrica especificada estiver abaixo do valor de destino. O ElastiCache for Redis aumenta a escala na horizontal de réplicas em um máximo de (% de desvio arredondado do objetivo, padrão 1) de réplicas existentes em todos os fragmentos no cluster.
- Uma política de escalabilidade de rastreamento de destino não escala quando a métrica especificada tem dados insuficientes. Ela não aumenta a escalabilidade porque não interpreta dados insuficientes como baixa utilização.
- É possível ver lacunas entre o valor de destino e os pontos de dados de métrica reais. Isso ocorre porque o Auto Scaling do ElastiCache for Redis funciona de maneira segura por arredondamento para cima ou para baixo, quando ele determina a capacidade a ser adicionada ou removida. Isso evita que ele adicione capacidade insuficiente ou remova muita capacidade.
- Para garantir a disponibilidade da aplicação, o serviço aumenta a escala na horizontal proporcionalmente à métrica o mais rápido possível, mas reduz a escala na horizontal mais gradualmente, com redução máxima de escala de 1 réplica ao longo dos fragmentos no cluster.
- Você pode ter várias políticas de escalabilidade de monitoramento do objetivo para um cluster do ElastiCache for Redis, desde que cada uma delas use uma métrica diferente. A intenção do Auto Scaling do ElastiCache for Redis é sempre priorizar a disponibilidade, portanto, seu comportamento será diferente dependendo se as políticas de monitoramento do objetivo estão prontas para aumentar ou reduzir a escala na horizontal. Ele vai aumentar o serviço se qualquer uma das políticas de monitoramento do objetivo estiverem prontas para aumentar, mas vai reduzir somente se todas as políticas de monitoramento do objetivo (com a parte de redução habilitada) estiverem prontas para reduzir.
- Não edite nem exclua os alarmes do CloudWatch que o Auto Scaling do serviço gerencia para uma política de escalabilidade com monitoramento do objetivo. O Auto Scaling do ElastiCache for Redis exclui os alarmes automaticamente quando você exclui a política de escalabilidade ou exclui o cluster.
- O Auto Scaling do ElastiCache for Redis não impede a modificação manual de réplicas nos fragmentos. Esses ajustes manuais não afetam alarmes existentes do CloudWatch que estejam conectados à política de escalabilidade, mas podem afetar métricas que possam acionar esses alarmes do CloudWatch.

- Esses alarmes do CloudWatch gerenciados pelo Auto Scaling são definidos na métrica do AVG em todos os fragmentos do cluster. Assim, ter fragmentos quentes pode resultar em qualquer cenário de:
 - escalabilidade quando não for necessário devido à carga em alguns fragmentos quentes que acionam um alarme do CloudWatch
 - não escalar quando necessário devido ao AVG agregado em todos os fragmentos que afetam o alarme não violarem.
- Os limites padrão do ElastiCache for Redis em nós por cluster ainda se aplicam. Então, ao optar pelo Auto Scaling, se você espera que os nós máximos sejam mais do que o limite padrão, solicite um aumento de limite em [Limites de serviço da AWS](#) e escolha o tipo de limite Nós por cluster por tipo de instância.
- Certifique-se de que você tenha ENIs (interfaces de rede elásticas) suficientes disponíveis em sua VPC, que são necessárias durante o aumento de escala na horizontal. Para obter mais informações, consulte [Interfaces de rede elástica](#).
- Se não houver capacidade suficiente disponível no EC2, o Auto Scaling do ElastiCache for Redis não aumentará a escala na horizontal até que a capacidade esteja disponível ou se você modificar manualmente o cluster para os tipos de instância que têm capacidade suficiente.
- O Auto Scaling do ElastiCache for Redis não oferece suporte à escalabilidade de réplicas com um cluster com um cluster tendo `ReservedMemoryPercent` menos que 25%. Para obter mais informações, consulte [Managing Reserved Memory](#) (Gerenciamento da memória reservada).

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 a uma ElastiCache para Redis

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.
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 utilização da CPU que você deseja manter nas ElastiCache réplicas. Esse valor deve ser ≥ 35 e ≤ 70 . As réplicas do cluster serão adicionadas ou removidas por manter a métrica próxima ao valor especificado.
10. (Opcional) Não há suporte do console para os períodos de desaquecimento de redução ou aumento de escala na horizontal. Use o AWS CLI para modificar os valores de resfriamento.
11. Em Capacidade mínima, digite o número mínimo de réplicas que a política ElastiCache de Auto Scaling do Redis precisa manter.
12. Em Capacidade máxima, digite o número máximo de réplicas que a política ElastiCache de Auto Scaling do Redis precisa manter. Esse valor deve ser ≥ 5 .
13. Escolha Create (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 a API Application Auto Scaling AWS CLI ou a API Application Auto Scaling. A primeira etapa é registrar seu ElastiCache grupo de replicação do Redis com ElastiCache o escalonamento automático do Redis.

Antes de usar o ElastiCache escalonamento automático do Redis com um ElastiCache cluster do Redis, você registra seu cluster com o escalonamento automático do ElastiCache Redis. Você faz isso para definir a dimensão e os limites de escalabilidade a serem aplicados a esse cluster. ElastiCache for Redis, o auto scaling dimensiona dinamicamente o cluster ElastiCache for Redis ao longo da dimensão `elasticache:replication-group:Replicas` escalável, que representa o número de réplicas de cluster por fragmento.

Uso da 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 ElastiCache for Redis, 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 pelo escalonamento automático do ElastiCache Redis. 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 pelo escalonamento automático do ElastiCache 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 ElastiCache para 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 ^  

```

Uso da 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 ElastiCache for Redis, por exemplo `replication-group/myscalablecluster`.
- **ScalableDimension** — Defina esse valor como `elasticache:replication-group:Replicas`.
- **MinCapacity** — O número mínimo de réplicas a serem gerenciadas ElastiCache pelo redimensionamento automático do 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 pelo redimensionamento automático do 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 ElastiCache para Redis chamado `myscalablecluster` com a API Application Auto Scaling. Este registro indica que o cluster deve ser escalado dinamicamente para ter de uma a 5 réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definir uma política de ajuste de escala

Uma configuração de política de escalabilidade de rastreamento de destino é representada por um bloco JSON no qual as métricas e valores de destino são definidos. Você pode salvar uma configuração de política de escalabilidade como um bloco JSON em um arquivo de texto. Você pode usar esse arquivo de texto ao invocar a AWS CLI ou a API do Application Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de política, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

As seguintes opções estão disponíveis para definir uma configuração de política de escalabilidade com monitoramento do objetivo:

Tópicos

- [Uso de uma métrica predefinida](#)
- [Editar uma política de escalabilidade](#)
- [Excluir uma política de escalabilidade](#)
- [Use AWS CloudFormation para políticas do Auto Scaling](#)
- [Escalabilidade programada](#)

Uso de uma métrica predefinida

Uma configuração de política de escalabilidade de rastreamento de destino é representada por um bloco JSON no qual as métricas e valores de destino são definidos. Você pode salvar uma configuração de política de escalabilidade como um bloco JSON em um arquivo de texto. Você pode usar esse arquivo de texto ao invocar a AWS CLI ou a API do Application Auto Scaling. Para obter mais informações sobre a sintaxe de configuração de política, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

As seguintes opções estão disponíveis para definir uma configuração de política de escalabilidade com monitoramento do objetivo:

Tópicos

- [Uso de uma métrica predefinida](#)
- [Uso de uma métrica personalizada](#)
- [Uso de períodos de desaquecimento](#)
- [Desabilitar a atividade de redução](#)
- [Aplicação de uma política de escalabilidade a um cluster do ElastiCache for Redis](#)

Uso de uma métrica predefinida

Com o uso de métricas predefinidas, você pode definir rapidamente uma política de escalabilidade com monitoramento do objetivo para um cluster do ElastiCache for Redis que funcione com o monitoramento do objetivo no Auto Scaling do ElastiCache for Redis. Atualmente, o ElastiCache for Redis dá suporte à seguinte métrica predefinida no ajuste de escala automático de réplicas do ElastiCache:

`ElastiCacheReplicaEngineCPUUtilization`: o valor médio da métrica `EngineCPUUtilization` no CloudWatch em todas as réplicas no cluster do ElastiCache for Redis. O valor médio da métrica `EngineCPUUtilization` no CloudWatch em todas as réplicas no cluster do ElastiCache for Redis. Você pode encontrar o valor da métrica agregada no CloudWatch em `ElastiCache for Redis ReplicationGroupId`, `Role` para `ReplicationGroupId` e réplica de função obrigatórios.

Para usar uma métrica predefinida em sua política de escalabilidade, crie uma configuração de rastreamento de destino para sua política de escalabilidade. Essa configuração deve incluir uma `PredefinedMetricSpecification` para a métrica predefinida e um `TargetValue` para o valor de destino dessa métrica.

Uso de uma métrica personalizada

Com o uso de métricas personalizadas, você pode definir uma política de escalabilidade de rastreamento de destino que atenda a suas exigências personalizadas. Você pode definir uma métrica personalizada com base em qualquer métrica do ElastiCache for Redis que mude na proporção da escalabilidade. Nem todas as métricas do ElastiCache for Redis funcionam para o monitoramento do objetivo. A métrica deve ser de utilização válida e descrever o quão ocupada uma instância está. O valor da métrica deve aumentar ou diminuir na proporção do número de réplicas no cluster. Essa aumento ou redução proporcional é necessário para usar os dados da métrica para expandir ou reduzir proporcionalmente o número de réplicas.

Example

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, uma métrica personalizada ajusta um cluster do ElastiCache for Redis com base na utilização média de uma CPU em 50 por cento 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 da API do Application Auto Scaling. O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica predefinida `ElastiCacheReplicaEngineCPUUtilization` é usada para ajustar um cluster do ElastiCache for Redis com base em uma utilização média da CPU de 40 por cento em todas as réplicas do 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 evitar que a configuração da política de escalabilidade com monitoramento do objetivo reduza a escala na horizontal do seu cluster do ElastiCache for Redis desabilitando a atividade de redução de escala na horizontal. Desabilitar a atividade de redução de escala na horizontal impede que a política de escalabilidade exclua réplicas, enquanto permite ao mesmo tempo que a política de escalabilidade adicione-as conforme necessário.

Você pode especificar um valor booleano para `DisableScaleIn` a fim de permitir ou evitar a atividade de redução de escala na horizontal no seu cluster. Para obter mais informações sobre `DisableScaleIn`, consulte [TargetTrackingScalingPolicyConfiguration](#) na Referência da API do Application Auto Scaling.

Example

O exemplo a seguir descreve uma configuração de rastreamento de destino para uma política de escalabilidade. Nessa configuração, a métrica predefinida `ElastiCacheReplicaEngineCPUUtilization` ajusta um cluster do ElastiCache for Redis com base em uma utilização média da CPU de 40 por cento em todas as réplicas do 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 do ElastiCache for Redis

Após registrar a escalabilidade automática do seu cluster do ElastiCache for Redis e definir uma política de escalabilidade, aplique a política de escalabilidade ao cluster registrado. Para aplicar uma política de escalabilidade a um cluster do ElastiCache for Redis, você pode usar a AWS CLI ou a API do Application Auto Scaling.

Como usar a AWS CLI

Para aplicar uma política de escalabilidade a seu cluster do ElastiCache for Redis, use o comando [put-scaling-policy](#) com os seguintes parâmetros:

- `--policy-name`: o nome da política de escalabilidade.
- `--policy-type`: defina este valor como `TargetTrackingScaling`.
- `--resource-id`: o identificador do recurso para o cluster do ElastiCache for Redis. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache for Redis, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:Replicas`.

- `--target-tracking-scaling-policy-configuration`: a configuração da política de escalabilidade com monitoramento do objetivo a ser usada para o cluster do ElastiCache for Redis.

Example

No exemplo a seguir, aplique a política de escalabilidade com monitoramento do objetivo chamada `miscalablepolicy` para um cluster do ElastiCache for Redis chamado `miscalablecluster` com a escalabilidade automática do ElastiCache for Redis. 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 miscalablepolicy \  
  --policy-type TargetTrackingScaling \  
  --resource-id replication-group/miscalablecluster \  
  --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 miscalablepolicy ^  
  --policy-type TargetTrackingScaling ^  
  --resource-id replication-group/miscalablecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:Replicas ^  
  --target-tracking-scaling-policy-configuration file://config.json
```

Uso da API

Para aplicar a política de escalabilidade em seu cluster do ElastiCache for Redis com a API do Application Auto Scaling, use a operação [PutScalingPolicy](#) da API do Application Auto Scaling com os seguintes parâmetros:

- **PolicyName:** o nome da política de escalabilidade.
- **PolicyType:** defina este valor como `TargetTrackingScaling`.
- **ResourceID:** o identificador do recurso para o cluster do ElastiCache for Redis. Para este parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é nome do cluster do ElastiCache for Redis, por exemplo `replication-group/myscalablecluster`.
- **ServiceNamespace:** defina este valor como `elasticache`.
- **ScalableDimension:** defina este valor como `elasticache:replication-group:Replicas`.
- **TargetTrackingScalingPolicyConfiguration:** a configuração da política de escalabilidade com monitoramento de objetivo a ser usada para o cluster do ElastiCache for Redis.

Example

No exemplo a seguir, aplique a política de escalabilidade com monitoramento do objetivo chamada `scalablepolicy` para um cluster do ElastiCache for Redis chamado `myscalablecluster` com a escalabilidade automática do ElastiCache for Redis. 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, a AWS CLI ou a API do Application Auto Scaling.

Editar 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 console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
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 Modify (Modificar).
8. Faça as alterações na política.
9. Escolha Modify (Modificar).

Edição de uma política de escalabilidade usando a AWS CLI ou a API do Application Auto Scaling

Você pode usar a AWS CLI ou a API do Application Auto Scaling para editar uma política de escalabilidade da mesma forma que você aplica uma política de escalabilidade:

- Ao usar a API do Application Auto Scaling, especifique o nome da política que você deseja editar no parâmetro `PolicyName`. Especifique novos valores para os parâmetros que você deseja alterar.

Para obter mais informações, consulte [Aplicação de uma política de escalabilidade a um cluster do ElastiCache for Redis](#).

Excluir uma política de escalabilidade

Você pode excluir uma política de escalabilidade usando o AWS Management Console, o AWS CLI ou a API Application Auto Scaling

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 Redis.
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 a AWS CLI ou a API Application Auto Scaling

Você pode usar a API Application Auto Scaling AWS CLI ou a Application Auto Scaling para excluir uma política de escalabilidade de um cluster. ElastiCache

CLI

Para excluir uma política de escalabilidade do seu cluster ElastiCache for 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 do ElastiCache Redis. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do ElastiCache cluster, por exemplo `replication-group/myscalablecluster`.
- `--service-namespace`: defina este valor como `elasticache`.
- `--scalable-dimension`: defina este valor como `elasticache:replication-group:Replicas`.

Example

No exemplo a seguir, a política de escalabilidade com monitoramento do objetivo chamada `myscalablepolicy` é excluída de um cluster do ElastiCache chamado `myscalablecluster`.

Para Linux, macOS ou Unix:

```
aws application-autoscaling delete-scaling-policy \
  --policy-name myscalablepolicy \
  --resource-id replication-group/myscalablecluster \
  --service-namespace elasticache \
  --scalable-dimension elasticache:replication-group:Replicas \
```

Para Windows:

```
aws application-autoscaling delete-scaling-policy ^
  --policy-name myscalablepolicy ^
  --resource-id replication-group/myscalablecluster ^
  --service-namespace elasticache ^
  --scalable-dimension elasticache:replication-group:Replicas ^
```

API

Para excluir uma política de escalabilidade do seu cluster ElastiCache for Redis, use a operação da API [DeleteScalingPolicy](#) Application Auto Scaling com os seguintes parâmetros:

- `PolicyName` — O nome da política de escalabilidade.
- `resourceID` — O identificador do recurso para ElastiCache o cluster do Redis. Para esse parâmetro, o tipo de recurso é `ReplicationGroup` e o identificador exclusivo é o nome do ElastiCache 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 de um ElastiCache cluster `myscalablepolicy` do Redis chamado `myscalablecluster` com a API Application Auto Scaling.

```
POST / HTTP/1.1
```

```
>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas"
}
```

Use AWS CloudFormation para políticas do Auto Scaling

Este trecho mostra como criar uma ação programada e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso [AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
```

```
ScalableDimension: 'elasticache:replication-group:Replicas'  
TargetTrackingScalingPolicyConfiguration:  
  PredefinedMetricSpecification:  
    PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization  
  TargetValue: 40
```

Escalabilidade programada

A escalabilidade com base em uma programação permite que você dimensione sua aplicação em resposta a alterações de demanda. Para usar a escalabilidade programada, crie ações programadas, que instruem o ElastiCache for Redis a executar ações de escalabilidade em momentos específicos. Quando cria uma ação programada, você especifica um cluster do ElastiCache for Redis existente, quando a ação de escalabilidade deve ocorrer, a capacidade mínima e a capacidade máxima. É possível criar ações programadas para escalar uma única vez ou de forma programada.

Você só pode criar uma ação programada para clusters do ElastiCache for 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 as seguintes APIs de application-autoscaling:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Use AWS CloudFormation para criar políticas do Auto Scaling

Este trecho mostra como criar uma ação programada e aplicá-la a um recurso [AWS::ElastiCache::ReplicationGroup](#) usando o recurso

[AWS::ApplicationAutoScaling::ScalableTarget](#). Ela usa as funções intrínsecas [Fn::Join](#) e [Ref](#) para construir a propriedade `ResourceId` com o nome lógico do recurso `AWS::ElastiCache::ReplicationGroup` especificado no mesmo modelo.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
    ScheduledActions:
      - EndTime: '2020-12-31T12:00:00.000Z'
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
          ScheduledActionName: First
          Schedule: 'cron(0 18 * * ? *)'
```

Modificar o modo de cluster

O Redis é um banco de dados distribuído na memória que oferece suporte à fragmentação e à replicação. Os clusters do ElastiCache para Redis são a implementação distribuída do Redis que permite que os dados sejam particionados em vários nós do Redis. Um cluster do ElastiCache para Redis tem dois modos de operação: o modo de cluster ativado (CME) e o modo de cluster desativado (CMD). No CME, o Redis funciona como um banco de dados distribuído com vários fragmentos e nós, enquanto no CMD, o Redis funciona como um único nó.

Antes de migrar do CMD para o CME, as seguintes condições devem ser atendidas:

Important

A configuração do modo de cluster só pode ser alterada do modo de cluster desativado para o modo cluster ativado. Não é possível reverter essa configuração.

- O cluster só pode ter chaves no banco de dados 0.
- Os aplicativos devem usar um cliente Redis que seja capaz de usar o protocolo Cluster e um endpoint de configuração.
- O failover automático deve ser ativado no cluster com no mínimo uma réplica.
- A versão mínima do mecanismo Redis necessária para a migração é 7.0.

Para migrar do CMD para o CME, a configuração do modo de cluster deve ser alterada do modo de cluster desativado para o modo cluster ativado. Esse é um procedimento de duas etapas que garante a disponibilidade do cluster durante o processo de migração.

Note

Você precisa fornecer um grupo de parâmetros com configuração habilitada para cluster, ou seja, o parâmetro habilitado para cluster deve estar definido como `yes`. Se você estiver usando um grupo de parâmetros padrão, o ElastiCache para Redis escolherá automaticamente o grupo de parâmetros padrão correspondente com uma configuração habilitada para cluster. O valor do parâmetro habilitado para cluster é definido como `no` para um cluster CMD. Conforme o cluster passa para o modo compatível, o valor do parâmetro habilitado para cluster é atualizado para `yes` como parte da ação de modificação. Para obter mais informações, consulte [Configuração de parâmetros do mecanismo usando grupos de parâmetros](#).

1. Preparar-se: crie um cluster CME de teste e certifique-se de que sua pilha esteja pronta para trabalhar com ele. O ElastiCache para Redis não tem como verificar sua prontidão. Para obter mais informações, consulte [Criar um cluster](#).
2. Modificar a configuração existente do cluster CMD para compatível com o modo cluster: nesse modo, haverá um único fragmento implantado e o ElastiCache para Redis 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 Redis Cluster e o endpoint de configuração. Para alterar o modo de cluster do 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 (exceto `cacheParameterGroupName`) não podem ser modificados ao definir o parâmetro do modo de cluster na solicitação [ModifyReplicationGroup](#).

- a. Usando o AWS Management Console, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Compatível
- b. Usando a API, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `ClusterMode` para `compatible`.
- c. Usando a AWS CLI, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `cluster-mode` para `compatible`.

Depois de alterar o modo de cluster do Redis para o modo de cluster compatível, a API [DescribeReplicationGroups](#) retornará o endpoint de configuração do cluster ElastiCache para Redis. 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 [Localização de endpoints de conexão](#).

3. Modificar a configuração do cluster para o modo cluster ativado: depois que o modo de cluster for definido como compatível com o modo de cluster, a segunda etapa é modificar a configuração do cluster para o modo cluster ativado. Nesse modo, um único fragmento está em execução, e os clientes agora podem ajustar a escala dos seus clusters ou modificar outras configurações de cluster.

Para alterar o modo de cluster para ativado, siga estas etapas:

Antes de começar, certifique-se de que seus clientes Redis tenham migrado para o protocolo de cluster e que o endpoint de configuração do cluster não esteja em uso.

- a. Usando o AWS Management Console, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Habilitado.
- b. Usando a API, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `ClusterMode` para `enabled`.

- c. Usando a AWS CLI, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `cluster-mode` para `enabled`.

Depois de alterar o modo de cluster para ativado, os endpoints serão configurados de acordo com a especificação do cluster Redis. A API [DescribeReplicationGroups](#) retornará o parâmetro do modo de cluster como `enabled` e os endpoints do cluster que agora estão disponíveis para serem usados pelos aplicativos para se conectar ao cluster.

Observe que os endpoints do cluster mudarão quando o modo de cluster for alterado para ativado. Certifique-se de atualizar seus aplicativos com os novos endpoints.

Você também pode optar por voltar ao modo de cluster desativado (CMD) usando o modo de cluster compatível e preservar as configurações originais.

Modificar a configuração do cluster para o modo de cluster desativado usando o modo de cluster compatível

1. Usando o AWS Management Console, consulte [Modificação de um grupo de replicação](#) e defina o modo de cluster como Desabilitado.
2. Usando a API, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `ClusterMode` para `disabled`.
3. Usando a AWS CLI, consulte [ModifyReplicationGroup](#) e atualize o parâmetro `cluster-mode` para `disabled`.

Depois de alterar o modo de cluster para desativado, a API [DescribeReplicationGroups](#) retornará o parâmetro do modo de cluster como `disabled`.

Replicação em regiões da AWS usando o armazenamentos de dados globais

Note

No momento, o Global Datastore só está disponível para clusters autoprojeados.

Usando o recurso Datastore global para Redis, você pode trabalhar com replicação totalmente gerenciada, rápida, confiável e segura em todas as regiões da AWS. Usando esse recurso, você pode criar clusters de réplicas de leitura entre regiões para o ElastiCache for Redis habilitar leituras de baixa latência e recuperação de desastres entre regiões da 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 \(console\)](#)
- [Uso de datastores 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 região da AWS diferente da região do cluster primário.

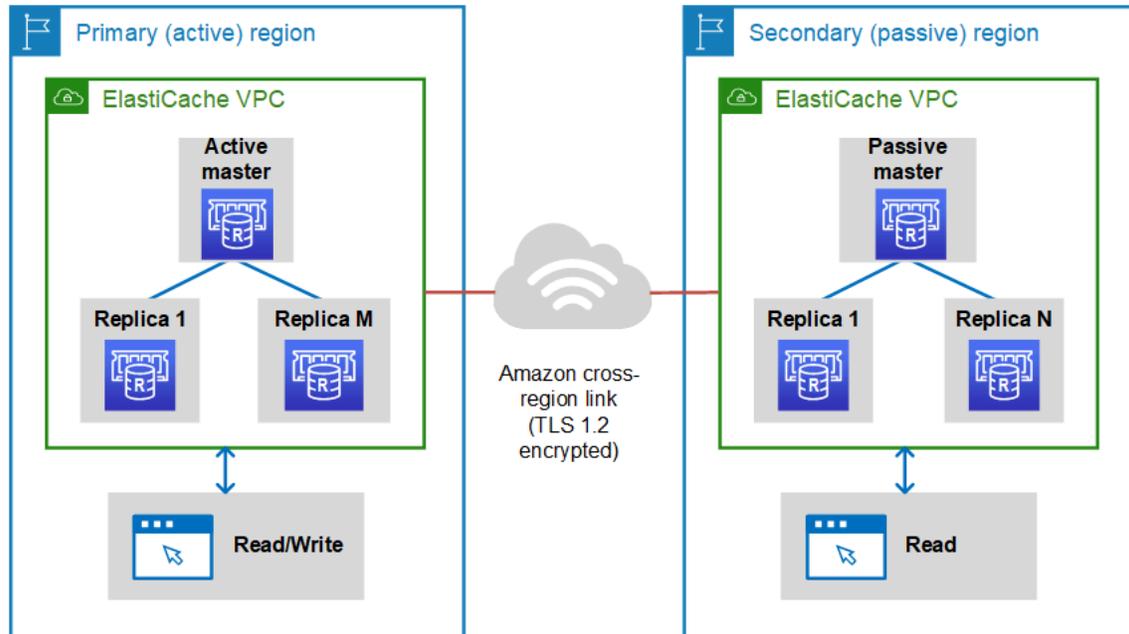
Quando você cria um datastore global no ElastiCache, o ElastiCache for Redis replica automaticamente os dados do cluster primário para o cluster secundário. Escolha a região da AWS em que os dados do Redis devem ser replicados e crie um cluster secundário nessa região da AWS. Depois, o ElastiCache configura e gerencia a replicação automática e assíncrona de dados entre os dois clusters.

O uso de um datastore global do Redis oferece as seguintes vantagens:

- Performance geolocal: ao configurar clusters de réplicas remotas em regiões adicionais da AWS e sincronizar seus dados entre eles, você pode reduzir a latência do acesso a dados nessa região da AWS. Um datastore global pode ajudar a aumentar a capacidade de resposta da sua aplicação, servindo leituras locais de baixa latência entre regiões da 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 região da AWS 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 não são compatíveis com instâncias anteriores a m5 ou r5.

- 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.

- É possível trabalhar com datastores globais somente em clusters de VPC. Para ter mais informações, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#). Os datastores globais não são compatíveis ao usar o EC2-Classic. Para obter mais informações, consulte [EC2-Classic](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

 Note

No momento, você não pode usar datastores globais no [Uso de zonas locais com o 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 ter mais informações, consulte [Dimensionamento ElastiCache para Redis](#).
- Os datastores globais oferecem suporte à [criptografia em repouso](#), à [criptografia em trânsito](#) e ao [AUTH do Redis](#).
- Os armazenamentos de dados globais não oferecem suporte ao Protocolo de Internet 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 keyspace, somente assinantes da mesma AWS região recebem os eventos.
 - Para eventos publicados no keyspace, 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 o Redis 5.0.6 ou posterior.
2. Adicione até dois clusters secundários em AWS regiões diferentes, novamente usando o mecanismo Redis 5.0.6 ou posterior.

Os procedimentos a seguir orientam você sobre como criar um armazenamento de dados global para o Redis e realizar outras operações usando o console do ElastiCache Redis.

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 o mecanismo do Redis 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 Redis e selecione um cluster.
3. Em Actions (Ações), escolha Setup Global Datastore (Configurar datastore global).
4. Na página Setup Global Datastore (Configurar datastore global) faça o seguinte:
 - Insira um valor para o Global Datastore Name suffix (Sufixo do nome do datastore global): esse sufixo é usado para gerar um nome exclusivo para o datastore global. Você pode pesquisar o datastore global usando o sufixo especificado aqui.
 - (Opcional) Insira um valor para Description (Descrição).
5. Em Secondary cluster details (Detalhes do cluster secundário), escolha uma região da AWS diferente em que o cluster será armazenado.
6. Em Redis settings (Configurações do Redis), insira um valor para Name (Nome) e, opcionalmente, para Description (Descrição) do cluster.
7. 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 a chave AWS KMS gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando chaves AWS KMS gerenciadas pelo cliente](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [Criptografia em trânsito](#). Para o mecanismo Redis versão 6.0 em diante, ao habilitar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Access Control (Controle de acesso):
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições.
 - 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 ter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
 - Redis AUTH Default User (Usuário padrão Redis AUTH): um mecanismo de autenticação para o servidor Redis. Para obter mais informações, consulte [Redis AUTH](#).
8. (Opcional) Conforme necessário, atualize as configurações restantes do cluster secundário. Elas são pré-preenchidas com os mesmos valores das do cluster primário, mas você pode atualizá-las para atender aos requisitos específicos para esse cluster.

- Porta
 - Número de réplicas
 - Grupo de sub-redes
 - Zona(s) de disponibilidade preferencial(is)
 - Grupos de segurança
 - Gerenciado pelo cliente (chave AWS KMS)
 - Token Redis AUTH
 - Habilitar backups automáticos
 - Backup retention period (Período de retenção de backup)
 - Janela de backup
 - Janela de manutenção
 - Tópico para notificação do SNS
9. 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 Redis é atualizada para indicar se um cluster faz parte de um datastore global, incluindo:

- Global Datastore (Datastore global): o nome datastore global ao qual o cluster pertence.
- Global Datastore Role (Função do datastore global): a função do cluster, primário ou secundário.

Você pode adicionar até um cluster secundário adicional em uma AWS região diferente. Para ter 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 Global Datastore (Datastore global) e escolha Create (Criar).
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 específicos do Redis](#) e [Criar um parameter group](#).

 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 ter 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 ter mais informações, consulte [Classificação de dados em níveis](#).

- e. Se você estiver criando um cluster do Redis (modo de cluster desabilitado):

Em Number of replicas (Número de réplicas), escolha o número de réplicas desejado para esse cluster.

- f. Se você estiver criando um cluster do Redis (modo cluster habilitado):

- i. Em Number of shards (Número de fragmentos), escolha o número de fragmentos (partições/grupos de nós) desejado para este cluster do Redis (modo cluster habilitado).

Para algumas versões do Redis (modo cluster habilitado), é possível alterar dinamicamente o número de fragmentos no cluster:

- Redis 3.2.10 e posterior: se o cluster estiver executando o Redis 3.2.10 ou versões posteriores, será possível alterar dinamicamente o número de fragmentos no cluster. Para ter mais informações, consulte [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#).
- Outras versões do Redis: se o cluster estiver executando uma versão do Redis 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 ter mais informações, consulte [Restauração de um backup para um novo cache](#).

- ii. Em Replicas per shard (Réplicas por fragmento), escolha o número de nós de réplica de leitura desejados em cada fragmento.

As restrições a seguir existem para o Redis (modo cluster habilitado).

- Se você tiver o Multi-AZ habilitado, verifique se tem pelo menos uma réplica por fragmento.
 - O número de réplicas é o mesmo para cada fragmento ao criar o cluster usando o console.
 - O número de réplicas de leitura por fragmento é fixo e não pode ser alterado. Se você achar que precisa de mais ou menos réplicas por fragmento (API/CLI: grupo de nós), deverá criar um novo cluster com o novo número de réplicas. Para ter mais informações, consulte [Propagação de um novo cluster autoprotetado com um backup criado externamente](#).
6. Em Configurações de grupo de sub-redes, escolha a sub-rede que você deseja aplicar a esse cluster. ElastiCache fornece um grupo de sub-rede IPv4 padrão ou você pode optar por criar um novo. Para o IPv6, é necessário criar um grupo de sub-rede com um bloco CIDR do IPv6. Se você escolher dual stack (pilha dupla), deverá selecionar um tipo de IP de descoberta: IPv6 ou IPv4.

Para obter mais informações, consulte [Criar uma sub-rede na sua VPC](#).

7. Em Availability zone placements (Posicionamentos de zona de disponibilidade), você tem duas opções:
- 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 ter mais informações, consulte [Escolher regiões e zonas de disponibilidade](#).

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 Advanced Redis settings (Configurações avançadas do Redis)

- Para Security (Segurança):

- Para criptografar seus dados, você tem as seguintes opções:

- Criptografia em repouso: permite a criptografia de dados armazenados em disco. Para obter mais informações, consulte [Criptografia em repouso](#).

Note

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

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [criptografia em trânsito](#). Para o mecanismo do Redis versão 6.0 em diante, ao habilitar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Access Control (Controle de acesso):

- No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.
- User Group Access Control List (Lista de controle de acesso de grupo de usuários): selecione um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para ter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
- Redis AUTH Default User (Usuário padrão Redis AUTH): um mecanismo de autenticação para o servidor Redis. Para obter mais informações, consulte [Redis AUTH](#).
- Redis AUTH (Redis AUTH: um mecanismo de autenticação para o servidor Redis. Para obter mais informações, consulte [Redis AUTH](#).

 Note

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

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

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

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

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

11. (Opcional) Especifique uma janela de manutenção. A opção janela de manutenção é o tempo, geralmente uma hora de duração, a cada semana quando o ElastiCache agenda a manutenção do sistema para seu cluster. Você pode 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 horários são em UCT.

Para ter mais informações, consulte [Gerenciamento da manutenção](#).

12. (Opcional) Para Logs:

- Em Log format (Formato do log), escolha Text (Texto) ou JSON.
- Em Tipo de destino, escolha CloudWatch Logs ou Kinesis Firehose.
- Em Destino do registro, escolha Criar novo e insira o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose, ou escolha Seleccionar existente e escolha o nome do grupo de CloudWatch registros do Logs ou o nome do stream do Firehose,

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 [Marcação de recursos do 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 a chave AWS KMS gerenciada pelo cliente e escolhendo a chave. Para obter mais informações, consulte [Usando chaves AWS KMS gerenciadas pelo cliente](#).

- Encryption in-transit (Criptografia em trânsito): permite a criptografia de dados na conexão. Para obter mais informações, consulte [Criptografia em trânsito](#). Para o mecanismo do Redis versão 6.4, se você habilitar a criptografia em trânsito, você será solicitado a especificar uma das seguintes opções de Access Control (Controle de acesso):
 - No Access Control (Sem controle de acesso): essa é a configuração padrão. Isso indica que não há restrições no acesso do usuário ao cluster.
 - User Group Access Control List (Lista de controle de acesso de grupo de usuários): escolha um grupo de usuários com um conjunto definido de usuários que possam acessar o cluster. Para ter mais informações, consulte [Gerenciamento de grupos de usuários com o console e a CLI](#).
 - Redis AUTH Default User (Usuário padrão Redis AUTH): um mecanismo de autenticação para o servidor Redis. Para obter mais informações, consulte [Redis AUTH](#).

 Note

Para versões do Redis entre 4.0.2, quando houve suporte para a criptografia em trânsito pela primeira vez, e 6.0.4, o Redis AUTH é a única opção.

As configurações de cluster secundário restantes são pré-preenchidas com os mesmos valores que o cluster primário, mas o seguinte pode ser atualizado para atender aos requisitos específicos para esse cluster:

- Porta
- Número de réplicas
- Grupo de sub-redes
- Zona(s) de disponibilidade preferencial(is)
- Grupos de segurança
- Gerenciado pelo cliente (chave AWS KMS)
- Token Redis AUTH
- Habilitar backups automáticos
- Backup retention period (Período de retenção de backup)
- Janela de backup
- Janela de manutenção
- Tópico para notificação do SNS

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 Redis também é atualizada para indicar se um cluster faz parte de um datastore global, incluindo o seguinte:

- Global Datastore (Datastore global): o nome datastore global ao qual o cluster pertence.
- Global Datastore Role (Função do datastore global): a função do cluster, primário ou secundário.

Você pode adicionar até um cluster secundário adicional em uma AWS região diferente. Para ter mais informações, consulte [Adição de uma região a um datastore global](#).

Visualização de detalhes do datastore global

Você pode visualizar os detalhes dos datastores globais existentes e também modificá-los na página Global Datastore (Datastore global).

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 Global Datastore (Datastore global) e escolha um datastore global disponível.

Depois, você pode examinar as seguintes propriedades do datastore global:

- Global Datastore Name: o nome do datastore global
- Descrição: uma descrição do datastore global
- Status: as opções incluem:
 - Criando
 - Modifying
 - Available (Disponível)
 - Excluindo
 - Somente primário - esse status indica que o datastore global contém apenas um cluster primário. Todos os clusters secundários são excluídos ou não são criados com êxito.
- Modo de cluster: habilitado ou desabilitado
- Versão do mecanismo do Redis: a versão do mecanismo do Redis que executa o datastore global
- Tipo de nó de instância: o tipo de nó usado para o datastore global
- Criptografia em repouso: habilitada ou desabilitada
- Criptografia em trânsito: Habilitada ou desabilitada
- Redis AUTH: habilitada ou desabilitada

Você pode fazer as seguintes alterações no datastore global:

- [Adição de uma região a um datastore global](#)
- [Remoção de uma região de um datastore global](#)

- [Promoção do cluster secundário para primário](#)
- [Modificação de um datastore global](#)

A página Global Datastore (Datastore global) também lista os clusters individuais que compõem o datastore global e as seguintes propriedades de cada um:

- 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 o Redis com modo cluster habilitado, o intervalo indica o atraso máximo 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 Global Datastore (Datastore global) e, em Global Datastore Name (Nome do datastore global), escolha um datastore global.
3. Escolha Adicionar região e escolha a AWS região em que o cluster secundário residirá.

4. Em Redis settings (Configurações do Redis), insira um valor para Name (Nome) e, opcionalmente, para Description (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 inativa

 Note

Você pode fornecer uma chave de criptografia diferente escolhendo a chave AWS KMS gerenciada pelo cliente e escolhendo a chave.

- Criptografia em trânsito
 - Redis AUTH
6. (Opcional) Atualize as configurações restantes do cluster secundário. Elas são pré-preenchidas com os mesmos valores das do cluster primário, mas é possível atualizá-las para atender aos requisitos específicos para esse cluster:
 - Porta
 - Número de réplicas
 - Grupo de sub-redes
 - **Zona(s) de disponibilidade preferencial(is)**

- Grupos de segurança
- (chave AWS KMS gerenciada pelo cliente)
- Token Redis AUTH
- Habilitar backups automáticos
- Backup retention period (Período de retenção de backup)
- Janela de backup
- Janela de manutenção
- Tópico para notificação do SNS

7. Escolha Add (Adicionar).

Modificação de um datastore global

É possível modificar propriedades de clusters regionais. Somente uma operação de modificação pode estar em andamento em um datastore global, com exceção da promoção de um cluster secundário para primário. Para ter 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 Global Datastore (Datastore global) e, em Global Datastore Name (Nome do datastore global), escolha um datastore global.
3. Escolha Modify (Modificar) e escolha entre as seguintes opções:
 - Modify description (Modificar descrição): atualiza a descrição do datastore global
 - Modify engine version (Modificar versão do mecanismo): somente o mecanismo do Redis versão 5.0.6 ou posterior está disponível.
 - 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 do Redis com modo de cluster habilitado:

- 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 console Redis ou a operação da [ModifyCacheClusterAPI](#). Para ter mais informações, consulte [Modificar um parameter group](#). Quando você modifica o grupo de parâmetros de qualquer cluster contido em um datastore global, ele é aplicado a todos os clusters dentro desse datastore global.

Para redefinir um grupo de parâmetros inteiro ou parâmetros específicos, use a operação de [ResetCacheParameterGroupAPI](#).

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 o Redis 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 Global Datastore (Datastore global) em Redis.
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 Global Datastore (Datastore global) em Redis.
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 ter 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 Global Datastore (Datastore global) em Redis.
3. Em Global Datastore Name (Nome do datastore global) escolha o datastore global desejado e escolha Delete (Excluir).

Depois, você será solicitado a confirmar sua decisão com o seguinte aviso: Are you sure you want to delete this Global Datastore?

4. Escolha Excluir.

O datastore global faz a transição para o status Deleting (Excluindo).

Uso de datastores globais (CLI)

Você pode usar a AWS Command Line Interface (AWS CLI) para controlar vários serviços da AWS via linha de comando e automatizá-los usando scripts. Você pode usar a CLI da AWS para operações ad hoc (únicas).

Download e configuração da AWS CLI

A AWS CLI é executada no Windows, no macOS ou no Linux. Use o procedimento a seguir para fazer download e configurá-la.

Como fazer download, instalar e configurar a CLI

1. Baixe a CLI da AWS na página [Interface de linha de comando da AWS](#).
2. Siga as instruções para instalação da CLI da AWS e configurar a CLI da AWS no Guia do usuário da AWS Command Line Interface.

Uso da CLI da AWS com datastores globais

Use as seguintes operações da CLI para trabalhar com datastores globais:

- [create-global-replication-group](#)

```
aws elasticache create-global-replication-group \  
  --global-replication-group-id-suffix my global datastore \  
  --primary-replication-group-id sample-repl-group \  
  --global-replication-group-description an optional description of the global  
  datastore
```

O Amazon ElastiCache aplica automaticamente um prefixo ao ID do datastore global quando ele é criado. Cada região da AWS tem seu próprio prefixo. Por exemplo, um ID de datastore global criado na região Oeste dos EUA (Norte da Califórnia) começa com "virxk" junto com o nome do sufixo fornecido. O sufixo, combinado com o prefixo gerado automaticamente, garante a exclusividade do nome do datastore global em várias regiões.

A tabela a seguir lista cada região da AWS e seu prefixo de ID de datastore global.

Nome da região/região	Prefixo
Região Leste dos EUA (Ohio) us-east-2	fpkhr
Região Leste dos EUA (N. da Virgínia)	ldgnf

Nome da região/região	Prefixo
us-east-1	
Região Leste dos EUA (Norte da Califórnia)	virxk
us-west-1	
Região Oeste dos EUA (Oregon)	sgau1
us-west-2	
Região do Canadá (Central)	bxodz
ca-central-1	
Região Ásia-Pacífico (Mumbai)	erpgt
ap-south-1	
Asia Pacific (Tokyo) Region	quwsw
ap-northeast-1	
Região Ásia-Pacífico (Seul)	lfqnh
ap-northeast-2	
Região Ásia-Pacífico (Osaka)	n1apn
ap-northeast-3	
Região Ásia-Pacífico (Singapura)	v1qxn
ap-southeast-1	
Região Ásia-Pacífico (Sydney)	vbgxd
ap-southeast-2	

Nome da região/região	Prefixo
Região Europa (Frankfurt) eu-central-1	iudkw
Região Europa (Irlanda) eu-west-1	gxeiz
Região Europa (Londres) eu-west-2	okuqm
Região Europa (Paris) eu-west-3	fgjhi
Região América do Sul (São Paulo) sa-east-1	juxlw
Região da China (Pequim) cn-north-1	emvgo
Região da China (Ningxia) cn-northwest-1	ckbem
Região Ásia-Pacífico (Hong Kong) ap-east-1	knjmp
AWS GovCloud (Oeste dos EUA) us-gov-west-1	sgwui

- [create-replication-group](#): use esta operação para criar clusters secundários para um datastore global fornecendo o nome do datastore global ao parâmetro `--global-replication-group-id`.

```
aws elasticache create-replication-group \  
  --replication-group-id secondary replication group name \  
  --replication-group-description "Replication group description" \  
  --global-replication-group-id global datastore name
```

Ao chamar esta operação e passar um valor `--global-replication-group-id`, o ElastiCache for Redis irá inferir os valores do grupo de replicação principal do grupo de replicação global para os seguintes parâmetros. Não passe valores para estes parâmetros:

"PrimaryClusterId",
"AutomaticFailoverEnabled",
"NumNodeGroups",
"CacheParameterGroupName",
"CacheNodeType",
"Engine",
"EngineVersion",
"CacheSecurityGroupNames",
"EnableTransitEncryption",
"AtRestEncryptionEnabled",
"SnapshotArns",
"SnapshotName"

- [describe-global-replication-groups](#)

```
aws elasticache describe-global-replication-groups \  
  --global-replication-group-id my global datastore \  
  --show-member-info an optional parameter that returns a list of the primary and  
  secondary clusters that make up the global datastore
```

- [modify-global-replication-group](#)

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --automatic-failover-enabled \  
  --cache-node-type node type \  
  --cache-parameter-group-name parameter group name \  
  --engine-version engine version \  
  --apply-immediately \  
  --global-replication-group-description description
```

- [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 \  
  --node-group-count 3
```

```
--global-replication-group-id global-replication-group-name \  
--node-group-count 3
```

- [rebalance-shards-in-global-replication-group](#)

```
aws elasticache rebalance-shards-in-global-replication-group \  
--apply-immediately yes \  
--global-replication-group-id global-replication-group-name
```

Use a ajuda para listar todos os comandos disponíveis do ElastiCache for Redis.

```
aws elasticache help
```

Você também pode usar a ajuda para descrever um comando específico e saber mais sobre seu uso:

```
aws elasticache create-global-replication-group help
```

Alta disponibilidade com o uso de grupos de replicação

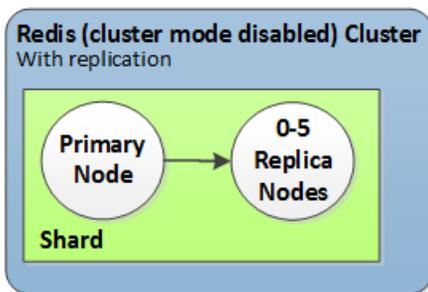
Os clusters do Redis do Amazon ElastiCache de nó único são entidades na memória com serviços de proteção de dados limitados (AOF). Se o seu cluster falhar por qualquer motivo, você perderá todos os dados do cluster. No entanto, se você estiver executando o mecanismo Redis, poderá agrupar de 2 a 6 nós em um cluster com réplicas em que 1 a 5 nós somente leitura contêm dados replicados do único nó primário de leitura/gravação do grupo. Nesse cenário, se um nó falhar por qualquer motivo, você não perderá todos os seus dados, pois eles serão replicados em um ou mais outros nós. Devido à latência de replicação, alguns dados podem ser perdidos se a falha ocorrer no nó primário de leitura/gravação.

Conforme visto no gráfico a seguir, a estrutura de replicação está contida dentro de um fragmento (chamado de grupo de nós na API/CLI) que está contido dentro de um cluster do Redis. Clusters do Redis (modo cluster desabilitado) sempre têm um fragmento. Clusters do Redis (modo cluster habilitado) podem ter até 500 fragmentos, com os dados do cluster particionados nos fragmentos. É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do 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 as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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



Cluster do Redis (modo cluster desabilitado) tem um fragmento e de 0 a 5 nós de réplicas

Se o cluster com réplicas tiver o Multi-AZ habilitado e o nó primário falhar, esse nó primário executará failover em uma réplica de leitura. Como os dados são atualizados nos nós de réplica de forma assíncrona, pode haver alguma perda de dados devido à latência na atualização dos nós de réplica. Para obter mais informações, consulte [Mitigar falhas ao executar o Redis](#).

Tópicos

- [Noções básicas sobre a replicação do Redis](#)
- [Replicação: Redis \(modo cluster desabilitado\) versus Redis \(modo cluster habilitado\)](#)
- [Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ](#)
- [Como a sincronização e o backup são implementados](#)
- [Criação de um grupo de replicação do Redis](#)
- [Visualização dos detalhes de um grupo de replicação](#)
- [Localização de endpoints de grupo de replicação](#)
- [Modificação de um grupo de replicação](#)
- [Exclusão de um grupo de replicação](#)

- [Alteração do número de réplicas](#)
- [Promoção de uma réplica de leitura a primário, para grupos de replicação do Redis \(modo cluster desabilitado\)](#)

Noções básicas sobre a replicação do Redis

O Redis implementa a replicação de duas formas:

- Com um único fragmento que contém todos os dados do cluster em cada nó: Redis (modo cluster desabilitado)
- Com dados particionados em até 500 fragmentos: Redis (modo cluster habilitado)

Cada fragmento em um grupo de replicação tem um único nó primário de leitura/gravação e até 5 nós de réplica somente de leitura. É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

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

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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

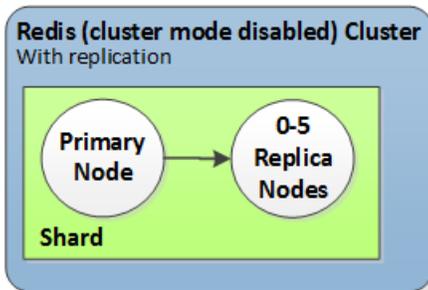
Tópicos

- [Redis \(modo cluster desabilitado\)](#)
- [Redis \(modo cluster habilitado\)](#)

Redis (modo cluster desabilitado)

Um cluster do Redis (modo cluster desabilitado) tem um único fragmento, dentro do qual há uma coleção de nós do Redis; um nó de leitura/gravação primário e até cinco nós de réplica secundários somente de leitura. Cada réplica de leitura mantém uma cópia dos dados do nó primário do cluster. Mecanismos de replicação assíncronos são usados para manter as réplicas de leitura sincronizadas com a primária. Os aplicativos podem ler a partir de qualquer nó no cluster. Os aplicativos podem

apenas gravar no nó primário. Réplicas de leitura melhoram o throughput de leitura e protegem contra a perda de dados nos casos de falha de um nó.



Cluster do Redis (modo cluster desabilitado) com um único fragmento e nós de réplicas

Você pode usar clusters do Redis (modo cluster desabilitado) com réplicas de nós para escalar sua solução Redis para o ElastiCache de modo a lidar com aplicações de leitura intensiva ou para oferecer suporte a um grande número de clientes que leiam simultaneamente do mesmo cluster.

Todos os nós em um cluster do Redis (modo cluster desabilitado) devem residir na mesma região.

Quando você adiciona uma réplica de leitura a um cluster, todos os dados do primário são copiados para o novo nó. A partir desse momento, sempre que os dados são gravados no primário, as alterações são propagadas de forma assíncrona em todas as réplicas de leitura.

Para melhorar a tolerância a falhas e reduzir o tempo de gravação, habilitar o recurso multi-AZ com failover automático para o seu cluster do Redis (modo cluster desabilitado) com réplicas. Para obter mais informações, consulte [Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ](#).

Você pode alterar as funções dos nós dentro do cluster do Redis (modo cluster desabilitado) com o primário e uma das réplicas trocando funções. Você pode decidir fazer isso por motivos de ajuste de desempenho. Por exemplo, com um aplicativo Web que possui uma intensa atividade de gravação, você pode escolher o nó que tenha a menor latência da rede. Para obter mais informações, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Redis \(modo cluster desabilitado\)](#).

Redis (modo cluster habilitado)

Um cluster do Redis (modo cluster habilitado) é composto de 1 a 500 fragmentos (API/CLI: grupos de nós). Cada fragmento tem um nó primário e até cinco nós de réplica de leitura. A configuração pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

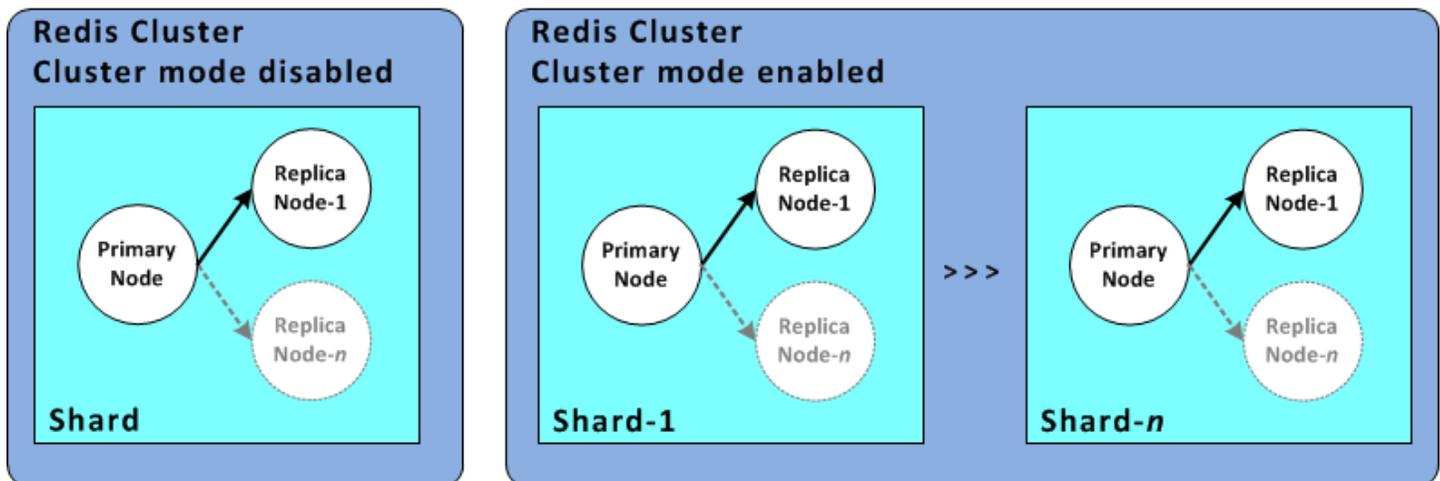
O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo Redis for 5.0.6 ou superior. Por exemplo, você pode optar por configurar um cluster de 500 nós que varia entre 83 fragmentos (uma primária e 5 réplicas por fragmento) e 500 fragmentos (primário único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

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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

Cada réplica de leitura em um fragmento mantém uma cópia dos dados do primário do fragmento. Mecanismos de replicação assíncronos são usados para manter as réplicas de leitura sincronizadas com a primária. Os aplicativos podem ler a partir de qualquer nó no cluster. Os aplicativos podem apenas gravar nos nós primários. Réplicas de leitura melhoram a escalabilidade de leitura e protegem contra a perda de dados. Os dados são particionados em todos os fragmentos em um cluster do Redis (modo cluster habilitado).

As aplicações usam o endpoint de configuração do cluster do Redis (modo cluster habilitado) para se conectar aos nós no cluster. Para obter mais informações, consulte [Localização de endpoints de conexão](#).



Cluster do Redis (modo cluster habilitado) com vários fragmentos e nós de réplica

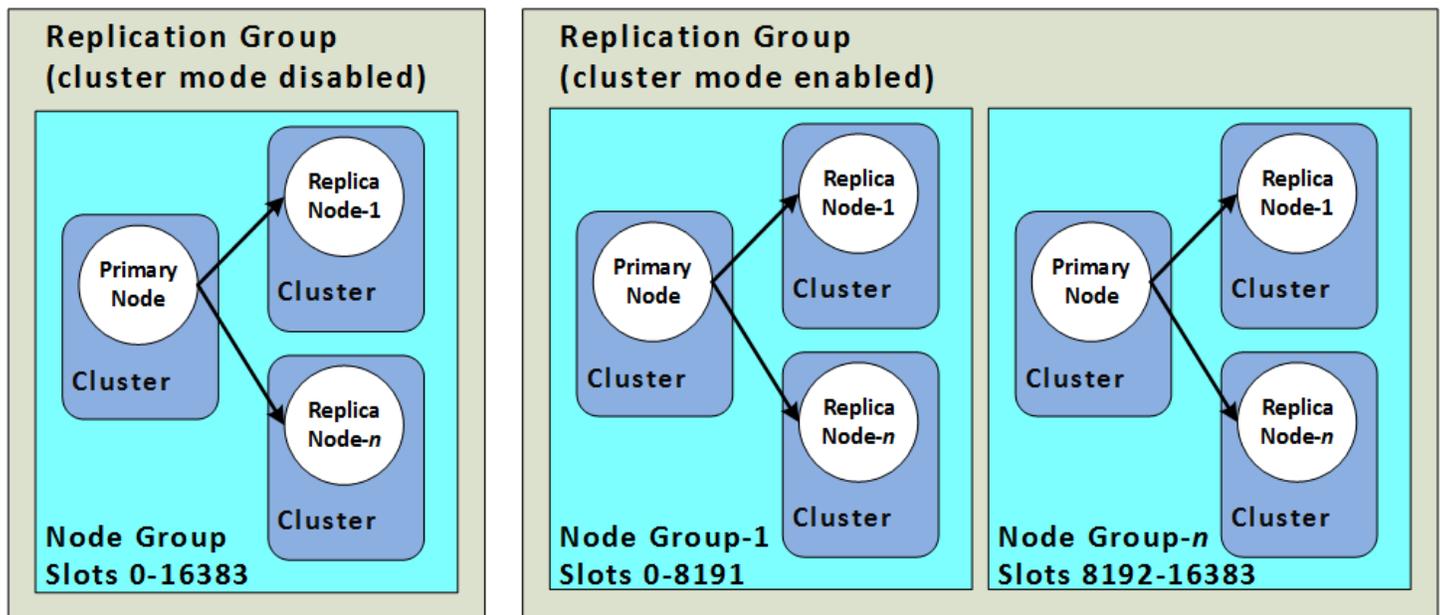
Todos os nós em um cluster do Redis (modo cluster habilitado) devem residir na mesma região. Para melhorar a tolerância a falhas, você pode provisionar primários e réplicas de leitura em várias zonas de disponibilidade dentro dessa região.

Atualmente, no Redis (modo cluster habilitado) existem algumas limitações.

- Você não pode promover manualmente nenhum nó de réplica para primário.

Replicação: Redis (modo cluster desabilitado) versus Redis (modo cluster habilitado)

Começando com o Redis versão 3.2, você pode criar um dos dois tipos distintos de clusters Redis (API/CLI: grupos de replicação). Um cluster do Redis (modo cluster desabilitado) sempre possui um único fragmento (API/CLI: grupo de nós) com até 5 nós de réplicas de leitura. Um cluster do Redis (modo cluster habilitado) tem até 500 fragmentos com de 1 a 5 nós de réplicas de leitura em cada.



Clusters do Redis (modo cluster desabilitado) e do Redis (modo cluster habilitado)

A tabela a seguir resume diferenças importantes entre os clusters do Redis (modo cluster desabilitado) e do Redis (modo cluster habilitado).

Comparando clusters do Redis (modo cluster desabilitado) e do Redis (modo cluster habilitado)

Recurso	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
Permite modificação	Sim. Oferece suporte para adicionar e excluir nós de réplica e dimensionar o tipo de nó.	Limitado. Para obter mais informações, consulte Versões de mecanismos e atualização e Escalabilidade de clusters

Recurso	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
		no Redis (modo cluster habilitado) .
Particionamento de dados	Não	Sim
Estilhaços	1	1 a 500
Réplicas de leitura	0 – 5  Important Se você não tiver réplicas e o nó falhar, ocorrerá uma perda total de dados.	0 a 5 por fragmento.  Important Se você não tiver réplicas e um nó falhar, ocorrerá uma perda de todos os dados nesse estilhaço.
Multi-AZ	Sim, com pelo menos 1 réplica. Opcional. Ativado por padrão.	Sim Opcional. Ativado por padrão.
Snapshots (Backups)	Sim, criando um único arquivo .rdb.	Sim, criando um arquivo .rdb exclusivo para cada estilhaço.
Restaurar	Sim, usando um único arquivo .rdb de um cluster do Redis (modo cluster desabilitado).	Sim, usando arquivos .rdb de um cluster do Redis (modo cluster desabilitado) ou do Redis (modo cluster habilitado).
Com suporte por	Todas as versões do Redis	Redis 3.2 e seguintes

Recurso	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
Mecanismo atualizável	Sim, com alguns limites. Para obter mais informações, consulte Versões de mecanismos e atualização .	Sim, com alguns limites. Para obter mais informações, consulte Versões de mecanismos e atualização .
Criptografia	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e 4.0.10 e posteriores.	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e 4.0.10 e posteriores.
Elegível para HIPAA	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e 4.0.10 e posteriores.	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e 4.0.10 e posteriores.
Compatível com PCI DSS	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e 4.0.10 e posteriores.	Versões 3.2.6 (programa das para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e 4.0.10 e posteriores.
Reestilhecimento online	N/D	Versão 3.2.10 (programa da para EOL, consulte Cronograma de fim de vida útil das versões do Redis) e posterior.

Qual devo escolher?

Ao escolher entre Redis (modo cluster desabilitado) ou Redis (modo cluster habilitado), considere os seguintes fatores:

- Escalabilidade versus particionamento: os negócios precisam mudar. Você precisa fornecer provisão para demanda máxima ou escalar conforme a demanda sofre alterações. O Redis (modo cluster desabilitado) oferece suporte a escalabilidade. Você pode dimensionar a capacidade de leitura adicionando ou excluindo nós de réplica ou pode dimensionar a capacidade aumentando para um tipo de nó maior. Ambas as operações exigem tempo. Para obter mais informações, consulte [Escalabilidade de clusters do Redis \(modo cluster desativado\) com nós de réplica](#).

O Redis (modo cluster habilitado) oferece suporte para o particionamento dos seus dados em até 500 grupos de nós. Você pode alterar dinamicamente o número de estilhaços à medida que suas necessidades de negócios mudam. Uma vantagem do particionamento é que você espalha sua carga em um número maior de endpoints que reduz o afunilamento de acesso durante a demanda de pico. Além disso, você pode acomodar um conjunto de dados maior, pois os dados podem ser distribuídos em vários servidores. Para obter informações sobre como escalar suas partições, consulte [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#).

- Tamanho do nó versus número de nós: como um cluster do Redis (modo cluster desabilitado) tem apenas um fragmento, o tipo de nó deve ser grande o suficiente para acomodar todos os dados do cluster mais a sobrecarga necessária. Por outro lado, como você pode particionar seus dados em vários fragmentos ao usar um cluster do Redis (modo cluster habilitado), os tipos de nós podem ser menores, embora você precise de mais deles. Para obter mais informações, consulte [Escolha do tamanho do nó](#).
- Leituras versus gravações: se a carga principal no seu cluster for de aplicações que leiam dados, você poderá escalar um cluster do Redis (modo cluster habilitado) adicionando e excluindo réplicas de leitura. No entanto, há um máximo de 5 réplicas de leitura. Se a carga no seu cluster for de escrita pesada, você poderá se beneficiar dos endpoints de gravação adicionais de um cluster do Redis (modo cluster habilitado) com vários fragmentos.

Seja qual for o tipo de cluster que você optar por implementar, certifique-se de escolher um tipo de nó que seja adequado para suas necessidades atuais e futuras.

Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ

Há várias instâncias em que o ElastiCache para Redis pode precisar substituir um nó primário; elas incluem certos tipos de manutenção planejada e o evento improvável de falha em um nó primário ou zona de disponibilidade.

Essa substituição resulta em algum tempo de inatividade do cluster, mas se o multi-AZ estiver habilitado, o tempo de inatividade será minimizado. A função do nó primário fará failover automaticamente para uma das réplicas de leitura. Não há necessidade de criar e provisionar um novo nó primário, porque o ElastiCache lidará com isso de forma transparente. O failover e a promoção de réplica garantem que você possa continuar a gravar no novo primário assim que a promoção estiver concluída.

O ElastiCache também propaga o nome do DNS (Serviço de Nomes de Domínio) da réplica promovida. Isso ocorre porque, em seguida, se o seu aplicativo estiver gravando no endpoint primário, nenhuma alteração nesse endpoint será necessária no seu aplicativo. Se estiver lendo de endpoints individuais, altere o endpoint de leitura da réplica promovida a primária para o endpoint da nova réplica.

No caso de substituições de nó planejadas, iniciadas devido a atualizações de manutenção ou atualizações de autoatendimento, esteja ciente do seguinte:

- Para cluster do ElastiCache para Redis, as substituições de nó planejadas agora são concluídas enquanto o cluster atende às solicitações de gravação recebidas.
- Para clusters desabilitados do modo de cluster do Redis com o Multi-AZ habilitado que são executados no mecanismo 5.0.6 ou posterior, as substituições de nó planejadas são concluídas enquanto o cluster atende às solicitações de gravação recebidas.
- Para clusters desabilitados no modo de cluster com Multi-AZ habilitado que são executados no mecanismo 4.0.10 ou anterior, é possível notar uma breve interrupção de gravação associada às atualizações de DNS. Essa interrupção pode levar até alguns segundos. Esse processo é muito mais rápido do que recriar e provisionar um novo primário, que será o caso se você não habilitar o multi-AZ.

É possível habilitar o multi-AZ usando o console de gerenciamento do ElastiCache, a AWS CLI ou a API do ElastiCache.

Habilitar o multi-AZ do ElastiCache no cluster do Redis (na API, na CLI ou no grupo de replicação) melhora a tolerância a falhas. Isso é verdade, especialmente nos casos em que o cluster primário

de leitura/gravação do seu cluster se torna inacessível ou falha por qualquer motivo. O Multi-AZ tem suporte em clusters do Redis que têm mais de um nó em cada fragmento.

Tópicos

- [Habilitar Multi-AZ](#)
- [Cenários de falha com respostas do multi-AZ](#)
- [Teste do failover automático](#)
- [Limitações no Multi-AZ do Redis](#)

Habilitar Multi-AZ

É possível habilitar o multi-AZ ao criar ou modificar um cluster (API, CLI ou grupo de replicação) usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

É possível habilitar o multi-AZ somente em clusters do Redis (modo cluster desabilitado) que têm pelo menos uma réplica de leitura disponível. Clusters sem réplicas de leitura não fornece alta disponibilidade ou tolerância a falhas. Para obter informações sobre como criar um cluster com replicação, consulte [Criação de um grupo de replicação do Redis](#). Para obter informações sobre como adicionar uma réplica de leitura a um cluster com replicação, consulte [Adicionando uma réplica de leitura, para grupos de replicação do Redis \(Modo cluster desabilitado\)](#).

Tópicos

- [Habilitação do multi-AZ \(console\)](#)
- [Habilitar o recurso multi-AZ \(AWS CLI\)](#)
- [Habilitação do multi-AZ \(API do ElastiCache\)](#)

Habilitação do multi-AZ (console)

É possível habilitar o multi-AZ usando o console do ElastiCache ao criar um novo cluster do Redis ou modificando um cluster do Redis existente com replicação.

O multi-AZ é habilitado por padrão em clusters do Redis (modo cluster habilitado).

⚠ Important

O ElastiCache habilitará automaticamente o multi-AZ somente se o cluster contiver pelo menos uma réplica em uma zona de disponibilidade diferente da principal em todos os fragmentos.

Habilitação do multi-AZ ao criar um cluster usando o console do ElastiCache

Para obter mais informações sobre esse processo, consulte [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#). Tenha uma ou mais réplicas e habilite o Multi-AZ.

Habilitação do multi-AZ em um cluster existente (console)

Para obter mais informações sobre esse processo, consulte Modificação de um cluster [Usando o AWS Management Console](#).

Habilitar o recurso multi-AZ (AWS CLI)

O exemplo de código a seguir usa a AWS CLI para habilitar o Multi-AZ para o grupo de replicação `redis12`.

⚠ Important

O grupo de replicação `redis12` já deve existir e ter pelo menos uma réplica de leitura disponível.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id redis12 \  
  --automatic-failover-enabled \  
  --multi-az-enabled \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id redis12 ^
```

```
--automatic-failover-enabled ^  
--multi-az-enabled ^  
--apply-immediately
```

A saída JSON desse comando deve ser semelhante ao seguinte.

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "One shard, two nodes",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2b",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis12-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis12-002"  
          }  
        ],  
        "NodeGroupId": "0001",  
        "PrimaryEndpoint": {  
          "Port": 6379,  
          "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"  
        }  
      }  
    ]  
  }  
}
```

```
    ],  
    "ReplicationGroupId": "redis12",  
    "SnapshotRetentionLimit": 1,  
    "AutomaticFailover": "enabling",  
    "MultiAZ": "enabled",  
    "SnapshotWindow": "07:00-08:00",  
    "SnapshottingClusterId": "redis12-002",  
    "MemberClusters": [  
        "redis12-001",  
        "redis12-002"  
    ],  
    "PendingModifiedValues": {}  
}  
}
```

Para obter mais informações, consulte estes tópicos na AWS CLI Referência de comandos:

- [create-cache-cluster](#)
- [create-replication-group](#)
- [modify-replication-group](#) na AWS CLI Referência de comandos.

Habilitação do multi-AZ (API do ElastiCache)

O exemplo de código a seguir usa a API do ElastiCache para habilitar o multi-AZ para o grupo de replicação `redis12`.

Note

Para usar esse exemplo, o grupo de replicação `redis12` já deve existir e ter pelo menos uma réplica de leitura disponível.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ApplyImmediately=true  
&AutoFailover=true  
&MultiAZEnabled=true  
&ReplicationGroupId=redis12  
&Version=2015-02-02  
&SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256  
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, veja estes tópicos na Referência da API do ElastiCache:

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Cenários de falha com respostas do multi-AZ

Antes da introdução do multi-AZ, o ElastiCache detectava e substituía os nós com falha de um cluster recriando e reprovisionando esses nós. Se você habilitar o Multi-AZ, um nó primário com falha fará failover para a réplica com o menor atraso de replicação. A réplica selecionada é promovida automaticamente para primário, o que é muito mais rápido do que criar e reprovisionar um novo nó primário. Esse processo normalmente demora apenas alguns segundos até que você possa gravar novamente no cluster.

Quando o multi-AZ está habilitado, o ElastiCache monitora continuamente o estado do nó primário. Se o nó primário falhar, uma das seguintes ações será realizada, dependendo do tipo da falha.

Tópicos

- [Cenários de falha quando somente o nó primário falha](#)
- [Cenários de falha quando o nó primário e algumas réplicas de leitura falham](#)
- [Cenários de falha quando cluster inteiro falha](#)

Cenários de falha quando somente o nó primário falha

Se somente o nó primário falhar, a réplica de leitura com o menor atraso de replicação será promovida a primária. Depois disso, uma réplica de leitura de substituição é criada e provisionada na mesma zona de disponibilidade que o primário com falha.

Quando somente o nó primário falha, o recurso multi-AZ do ElastiCache faz o seguinte:

1. O nó primário com falha é colocado offline.
2. A réplica de leitura com o menor atraso de replicação é promovida a primário.

As gravações poderão ser retomadas assim que o processo de promoção estiver concluído, normalmente depois de apenas alguns segundos. Se a sua aplicação estiver gravando no endpoint primário, não será necessário alterar o endpoint para gravações ou leituras. O ElastiCache propaga o nome do DNS da réplica promovida.

3. Uma réplica de leitura de substituição é executada e provisionada.

A réplica de leitura de substituição é executada na Zona de disponibilidade em que o nó primário com falha se encontrava, para que a distribuição de nós seja mantida.

4. As réplicas são sincronizadas com o novo nó primário.

Depois que a nova réplica estiver disponível, lembre-se dos seguintes efeitos:

- Endpoint primário: não faça alterações na sua aplicação, pois o nome do DNS do novo nó primário é propagado para o endpoint primário.
- Endpoint leitor: o endpoint leitor é atualizado automaticamente para apontar para os novos nós de réplica.

Para obter informações sobre como encontrar os endpoints de um cluster, consulte os seguintes tópicos:

- [Localização de endpoints de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Encontrar os endpoints para grupos de replicação \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação \(API do ElastiCache\)](#)

Cenários de falha quando o nó primário e algumas réplicas de leitura falham

Se o primário e pelo menos uma réplica de leitura falhar, a réplica disponível com o menor atraso de replicação será promovida a cluster primário. Novas réplicas de leitura também são criadas e provisionadas nas mesmas Zonas de disponibilidade que os nós com falha e a réplica que foi promovida a primário.

Quando o nó primário e algumas réplicas de leitura falham, o multi-AZ do ElastiCache faz o seguinte:

1. O nó primário com falha e as réplicas de leitura com falha são colocadas offline.
2. A réplica disponível com o menor atraso de replicação é promovida a nó primário.

As gravações poderão ser retomadas assim que o processo de promoção estiver concluído, normalmente depois de apenas alguns segundos. Se a sua aplicação estiver gravando no endpoint primário, não será necessário alterar o endpoint para gravações. O ElastiCache propaga o nome do DNS da réplica promovida.

3. Réplicas de substituição são criadas e provisionadas.

As réplicas de substituição são criadas nas Zonas de disponibilidade dos nós com falha, de modo que a distribuição de nós seja mantida.

4. Todos os clusters são sincronizados com o novo nó primário.

Faça as seguintes alterações no seu aplicativo depois que os novos nós estiverem disponíveis:

- Endpoint primário: não faça alterações em sua aplicação. O nome de DNS do novo nó primário é propagado para o endpoint primário.
- Endpoint leitor: o endpoint leitor é atualizado automaticamente para apontar para os novos nós de réplica.

Para obter informações sobre como encontrar os endpoints de um grupo de replicação, consulte os seguintes tópicos:

- [Localização de endpoints de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Encontrar os endpoints para grupos de replicação \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação \(API do ElastiCache\)](#)

Cenários de falha quando cluster inteiro falha

Se tudo falhar, todos os nós serão recriados e provisionados nas mesmas Zonas de disponibilidade que os nós originais.

Nesse cenário, todos os dados do cluster são perdidos devido à falha de cada nó no cluster. Essa ocorrência é rara.

Quando o cluster inteiro falha, o multi-AZ do ElastiCache faz o seguinte:

1. O nó primário e as réplicas de leitura com falha são colocados offline.
2. Um nó primário de substituição é criado e provisionado.
3. Réplicas de substituição são criadas e provisionadas.

As substituições são criadas nas Zonas de disponibilidade dos nós com falha, de modo que a distribuição de nós seja mantida.

Como o cluster inteiro falhou, os dados são perdidos, e todos os novos nós são iniciados a frio.

Como cada um dos nós de substituição tem o mesmo endpoint que o nó que ele está substituindo, não é necessário fazer alterações de endpoint no seu aplicativo.

Para obter informações sobre como encontrar os endpoints de um grupo de replicação, consulte os seguintes tópicos:

- [Localização de endpoints de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Encontrar os endpoints para grupos de replicação \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação \(API do ElastiCache\)](#)

Recomendamos que você crie o nó primário e as réplicas de leitura em diferentes Zonas de disponibilidade para aumentar o nível de tolerância a falhas.

Teste do failover automático

Depois de habilitar o failover automático, é possível testá-lo usando o console do ElastiCache, a AWS CLI e a API do ElastiCache.

Ao testar, observe o seguinte:

- Você pode usar essa operação para testar o failover automático em até cinco fragmentos (chamados de grupos de nós na API do ElastiCache e na AWS CLI) em qualquer período contínuo de 24 horas.
- Se você chamar essa operação em estilhaços em clusters diferentes (chamados grupos de replicação na API e CLI), poderá fazer as chamadas simultaneamente.
- Em alguns casos, é possível chamar essa operação várias vezes em diferentes fragmentos no mesmo grupo de replicação do Redis (modo cluster habilitado). Nesses casos, a substituição do primeiro nó deve ser concluída antes que uma chamada subsequente possa ser feita.
- Para determinar se a substituição do nó está completa, verifique os eventos usando o console do Amazon ElastiCache, a AWS CLI ou a API do ElastiCache. Procure pelos seguintes eventos relacionados ao failover automático, listados aqui em ordem de ocorrência:
 1. Mensagem do grupo de replicação: `Test Failover API called for node group <node-group-id>`
 2. Mensagem do cluster de cache: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 3. Mensagem do grupo de replicação: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 4. Mensagem do cluster de cache: `Recovering cache nodes <node-id>`
 5. Mensagem do cluster de cache: `Finished recovery for cache nodes <node-id>`

Para obter mais informações, consulte as informações a seguir.

- [Visualização de eventos do ElastiCache](#) no Guia do usuário do ElastiCache
- [DescribeEvents](#) na Referência da API do ElastiCache
- [describe-events](#) na AWS CLI Referência de comandos.
- Essa API foi projetada para testar o comportamento de sua aplicação em caso de failover do ElastiCache. Ela não foi projetado para ser uma ferramenta operacional para iniciar um failover a fim de resolver um problema com o cluster. Além disso, sob determinadas condições, como eventos operacionais de grande escala, a AWS pode bloquear essa API.

Tópicos

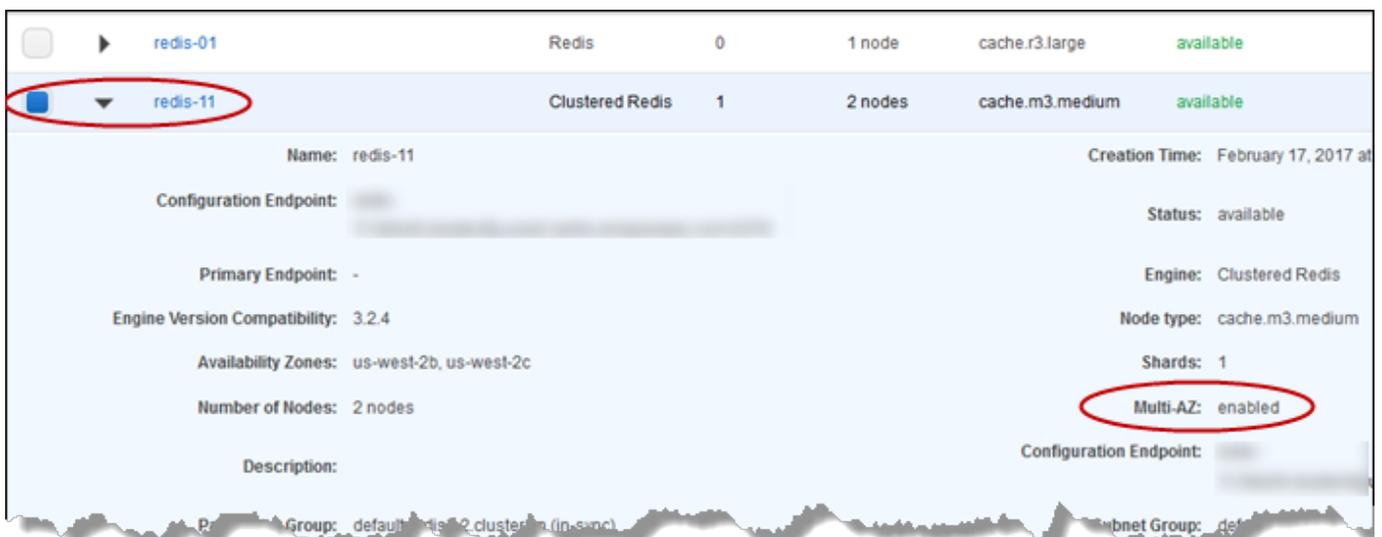
- [Teste do failover automático usando o AWS Management Console](#)
- [Teste do failover automático usando o AWS CLI](#)
- [Teste do failover automático usando a API do ElastiCache](#)

Teste do 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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis.
3. Na lista de clusters Redis, escolha a caixa à esquerda do cluster que deseja testar. Esse cluster deve ter pelo menos um nó de réplica de leitura.
4. Na área Details, confirme se esse cluster está habilitado para Multi-AZ. Se o cluster não estiver habilitado para o Multi-AZ, escolha um cluster diferente ou modifique esse cluster para habilitar o Multi-AZ. Para obter mais informações, consulte [Usando o AWS Management Console](#).



5. Para o Redis (modo cluster desabilitado), escolha o nome do cluster.

Para o Redis (modo cluster habilitado), faça o seguinte:

- a. Escolha o nome do cluster.

- b. Na página Shards, para o estilhaço (chamado de grupo de nó na API e na CLI) no qual você deseja testar o failover, escolha o nome do estilhaço.
6. Na página Nodes, escolha Failover Primary.
7. Escolha Continue para fazer failover do primário ou Cancel para cancelar a operação e não fazer failover do nó primário.

Durante o processo de failover, o console continua a mostrar o status do nó como disponível. Para acompanhar o progresso do seu teste de failover, escolha Events no painel de navegação do console. Na guia Events, observe os eventos que indicam que o failover foi iniciado (`Test Failover API called`) e concluído (`Recovery completed`).

Teste do failover automático usando o AWS CLI

É possível testar o failover automático em qualquer cluster habilitado para Multi-AZ usando a operação da AWS CLI `test-failover`.

Parâmetros

- `--replication-group-id`: obrigatório. O grupo de replicação (no console, cluster) que deve ser testado.
- `--node-group-id`: obrigatório. O nome do grupo de nós nos quais você deseja testar o failover automático. Você pode testar um máximo de cinco grupos de nós em um período contínuo de 24 horas.

O exemplo a seguir usa a AWS CLI para testar o failover automático no grupo de nós `redis00-0003` do cluster do Redis (modo cluster habilitado) `redis00`.

Exemplo Teste de failover automático

Para Linux, macOS ou Unix:

```
aws elasticache test-failover \  
  --replication-group-id redis00 \  
  --node-group-id redis00-0003
```

Para Windows:

```
aws elasticache test-failover ^  
--replication-group-id redis00 ^  
--node-group-id redis00-0003
```

A saída do comando precedente é semelhante ao seguinte.

```
{  
  "ReplicationGroup": {  
    "Status": "available",  
    "Description": "1 shard, 3 nodes (1 + 2 replicas)",  
    "NodeGroups": [  
      {  
        "Status": "available",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2c",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-002"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2b",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,
```

```
        "Address":
"redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"
      },
      "CacheClusterId": "redis1x3-003"
    }
  ],
  "NodeGroupId": "0001",
  "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
  }
}
],
"ClusterEnabled": false,
"ReplicationGroupId": "redis1x3",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotWindow": "11:30-12:30",
"SnapshottingClusterId": "redis1x3-002",
"MemberClusters": [
  "redis1x3-001",
  "redis1x3-002",
  "redis1x3-003"
],
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled",
"PendingModifiedValues": {}
}
}
```

Para acompanhar o progresso do failover, use a operação da AWS CLI `describe-events`.

Para obter mais informações, consulte as informações a seguir.

- [test-failover](#) na AWS CLI Referência de comandos.
- [describe-events](#) na AWS CLI Referência de comandos.

Teste do failover automático usando a API do ElastiCache

É possível testar o failover automático em qualquer cluster habilitado com multi-AZ usando a operação de API do ElastiCache `TestFailover`.

Parâmetros

- `ReplicationGroupId`: obrigatório. O grupo de replicação (no console ou cluster) a ser testado.
- `NodeGroupId`: obrigatório. O nome do grupo de nós nos quais você deseja testar o failover automático. Você pode testar um máximo de cinco grupos de nós em um período contínuo de 24 horas.

O exemplo a seguir testa o failover automático no grupo de nós `redis00-0003` no grupo de replicação (no console, cluster) `redis00`.

Exemplo Teste do failover automático

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=TestFailover  
&NodeGroupId=redis00-0003  
&ReplicationGroupId=redis00  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Para acompanhar o progresso do failover, use a operação de API do ElastiCache `DescribeEvents`.

Para obter mais informações, consulte as informações a seguir.

- [TestFailover](#) na Referência da API do ElastiCache
- [DescribeEvents](#) na Referência da API do ElastiCache

Limitações no Multi-AZ do Redis

Esteja ciente das seguintes limitações para o Multi-AZ do Redis:

- O Multi-AZ tem suporte no Redis versão 2.8.6 e posterior.
- O Multi-AZ do Redis não tem suporte em tipos de nó T1.
- A replicação 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, o ElastiCache for Redis escolhe a réplica com o menor atraso de replicação. Em outras palavras, ele escolhe a réplica que é mais atual. Isso ajuda a minimizar a quantidade de dados perdidos. A réplica com o atraso de replicação mínimo pode estar na mesma zona de disponibilidade ou em outra em relação ao nó primário com falha.

- Ao promover manualmente as réplicas de leitura para primárias no Redis (modo cluster desabilitado), é possível fazê-lo somente quando o multi-AZ e o failover automático estiverem desabilitados. Para promover uma réplica de leitura para primário, execute as seguintes etapas:
 1. Desabilite o Multi-AZ no cluster.
 2. Desabilite o failover automático no cluster. Isso pode ser feito usando o console do Redis desmarcando a caixa de seleção Auto failover (Failover automático) para o grupo de replicação. É possível fazer isso usando o AWS CLI definindo a propriedade `AutomaticFailoverEnabled` como `false` ao executar a operação `ModifyReplicationGroup`.
 3. Promova a réplica de leitura para primário.
 4. Habilite novamente o Multi-AZ.
- O multi-AZ do ElastiCache for Redis e o arquivo somente de acréscimo (AOF) são mutuamente exclusivos. Se você habilitar um, não é possível habilitar o outro.
- Uma falha de um nó pode ser causada pelo evento raro de falha total de uma zona de disponibilidade. Neste caso, a réplica que substitui o primário com falha é criada somente quando a zona de disponibilidade volta a ficar ativa. Por exemplo, considere um grupo de replicação com o primário em AZ-a e réplicas em AZ-b e AZ-c. Se o primário falhar, a réplica com o menor atraso de replicação será promovida a cluster primário. O ElastiCache criará uma nova réplica em AZ-a (onde o primário com falha estava localizado) somente quando AZ-a voltar a ficar ativa e disponível.
- Uma reinicialização iniciada pelo cliente de um primário não aciona o failover automático. Outras reinicializações e falhas desencadeiam o failover automático.
- Quando o primário é reiniciado, seus dados são limpos quando ele volta a ficar online. Quando as réplicas de leitura veem o cluster primário limpo, elas limpam suas cópias dos dados, o que causa perda de dados.

- Depois que uma réplica de leitura foi promovida, as outras réplicas se sincronizam com o novo primário. Após a sincronização inicial, o conteúdo das réplicas é excluído, e eles sincronizam os dados do novo primário. Esse processo de sincronização causa uma breve interrupção, durante a qual as réplicas não são acessíveis. O processo de sincronização também causa um aumento de carga temporário no primário durante a sincronização com as réplicas. Esse comportamento é nativo do Redis e não é exclusivo do multi-AZ do ElastiCache. Para obter detalhes sobre esse comportamento do Redis, consulte [Replication](#) no site do Redis.

 Important

Para o Redis versão 2.8.22 e posterior, não é possível criar réplicas externas.

Para versões do Redis anteriores à 2.8.22, recomendamos não conectar uma réplica externa do Redis a um cluster do ElastiCache for Redis habilitado para multi-AZ. Essa configuração sem suporte pode criar problemas que impedem que o ElastiCache realize corretamente o processo de failover e recuperação. Para conectar uma réplica do Redis externa a um cluster do ElastiCache, verifique se o multi-AZ não está habilitado antes de fazer a conexão.

Como a sincronização e o backup são implementados

Todas as versões com suporte do Redis oferecem suporte para backup e sincronização entre os nós primário e de réplica. No entanto, a forma como o backup e a sincronização são implementados varia de acordo com a versão Redis.

Redis versão 2.8.22 e posterior

Para a replicação Redis, nas versões 2.8.22 e posteriores, escolha um dos dois métodos de backup. Para obter mais informações, consulte [Versões do Redis anteriores a 2.8.22](#) e [Snapshots e restauração](#).

Durante o processo sem permissão, se as cargas de gravação forem pesadas, as gravações no cluster serão atrasadas para garantir que você não acumule muitas mudanças e, assim, evite um snapshot bem-sucedido.

Versões do Redis anteriores a 2.8.22

O backup e a sincronização do Redis nas versões anteriores a 2.8.22 é um processo de três etapas.

1. A bifurcação e, no processo em segundo plano, serializam os dados do cluster no disco. Isso cria um snapshot de ponto no tempo.
2. Em primeiro plano, acumule um log de alterações no buffer de saída do cliente.

Important

Se o log de alterações exceder o tamanho do buffer de saída do cliente, o backup ou a sincronização falhará. Para obter mais informações, consulte [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#).

3. Por fim, transmita os dados do cache e, em seguida, o log de alterações para o nó de réplica.

Criação de um grupo de replicação do Redis

Você tem as seguintes opções para criar um cluster com nós de réplica. Se aplica quando você já tem um cluster do Redis (modo cluster desabilitado) disponível não associado a nenhum cluster que possua réplicas a serem usadas como o nó primário. A outra é aplicada quando você precisa criar um nó primário com o cluster e as réplicas de leitura. Atualmente, um cluster do Redis (modo cluster habilitado) deve ser criado do zero.

opção 1: [Criação de um grupo de replicação usando um cluster do Redis \(modo cluster desabilitado\) disponível](#)

Use essa opção para aproveitar um cluster do Redis (modo cluster desabilitado) de nó único. 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 do Redis \(modo cluster desabilitado\) disponível](#).

Important

Não é possível criar um cluster do Redis (modo cluster habilitado) usando um cluster existente. Para criar um cluster do Redis (modo cluster habilitado) (API/CLI: grupo de replicação) usando o console do ElastiCache, consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#).

opção 2: [Criação de um grupo de replicação do Redis do princípio](#)

Use essa opção se você ainda não tiver um cluster do Redis (modo cluster desabilitado) disponível a ser usado como nó primário do cluster, ou se quiser criar um cluster do Redis (modo cluster habilitado). Consulte [Criação de um grupo de replicação do Redis do princípio](#).

Criação de um grupo de replicação usando um cluster do Redis (modo cluster desabilitado) disponível

Um cluster disponível é um cluster Redis existente de um único nó. Atualmente, o cluster do Redis (modo cluster habilitado) não oferece suporte para criar um cluster com réplicas usando um cluster de um único nó disponível. Se quiser criar um cluster do Redis (modo cluster habilitado), consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#).

O procedimento a seguir só pode ser usado se você tiver um cluster do Redis (modo cluster desabilitado) de um único nó. O nó desse cluster torna-se o primário no novo cluster. Se você não tiver um cluster do Redis (modo cluster desabilitado) que possa usar como primário do novo cluster, consulte [Criação de um grupo de replicação do Redis do princípio](#).

Criação de um grupo de replicação usando um cluster do Redis disponível (console)

Consulte o tópico [Usar a AWS Management Console](#).

Criação de um grupo de replicação usando um cluster de cache Redis disponível (AWS CLI)

Existem duas etapas para criar um grupo de replicação com réplicas de leitura ao usar um cluster de cache Redis disponível para o primário ao usar a AWS CLI.

Ao usar a AWS CLI você cria um grupo de replicação especificando o nó autônomo disponível como o nó primário do cluster, `--primary-cluster-id` e o número de nós que deseja no cluster usando o comando da CLI, `create-replication-group`. Inclua os seguintes parâmetros.

`--replication-group-id`

O nome do grupo de replicação que você está criando. O valor desse parâmetro é usado como a base para os nomes dos nós adicionados com um número sequencial de 3 dígitos acrescentados ao final de `--replication-group-id`. Por exemplo, `sample-repl-group-001`.

As restrições de nomenclatura do grupo de replicação do Redis (modo cluster desabilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

--replication-group-description

Descrição do grupo de replicação.

--num-node-groups

O número de nós desejados neste cluster. Esse valor inclui o nó primário. Esse parâmetro tem um valor máximo de seis.

--primary-cluster-id

O nome do cluster do Redis (modo cluster desabilitado) disponível que você deseja que seja o nó primário nesse grupo de replicação.

O comando a seguir cria o grupo de replicação `sample-repl-group` usando o cluster do Redis (modo cluster desabilitado) disponível `redis01` como o nó primário do grupo de replicação. Ele cria 2 novos nós que são réplicas de leitura. As configurações de `redis01` (ou seja, grupo de parâmetros, grupo de segurança, tipo de nó, versão do mecanismo, etc.) serão aplicadas a todos os nós no grupo de replicações.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --primary-cluster-id redis01
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a AWS CLI [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 mais informações, consulte os tópicos da AWS CLI:

- [create-replication-group](#)
- [modify-replication-group](#)

Adição de réplicas a um cluster independente do Redis (modo cluster desabilitado) (API do ElastiCache)

Ao usar a API do ElastiCache, você cria um grupo de replicação especificando o nó autônomo disponível como o nó primário do cluster, `PrimaryClusterId` e o número de nós que deseja no cluster usando o comando da CLI, `CreateReplicationGroup`. Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando. O valor desse parâmetro é usado como a base para os nomes dos nós adicionados com um número sequencial de 3 dígitos acrescentados ao final de `ReplicationGroupId`. Por exemplo, `sample-repl-group-001`.

As restrições de nomenclatura do grupo de replicação do Redis (modo cluster desabilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

ReplicationGroupDescription

Descrição do cluster com réplicas.

NumCacheClusters

O número de nós desejados neste cluster. Esse valor inclui o nó primário. Esse parâmetro tem um valor máximo de seis.

PrimaryClusterId

O nome do cluster do Redis (modo cluster desabilitado) disponível que você deseja que seja o nó primário nesse cluster.

O comando a seguir cria o cluster com réplicas `sample-repl-group` usando o cluster do Redis (modo cluster desabilitado) disponível `redis01` como o nó primário do grupo de replicação. Ele cria 2 novos nós que são réplicas de leitura. As configurações de `redis01` (ou seja, grupo de parâmetros, grupo de segurança, tipo de nó, versão do mecanismo, etc.) serão aplicadas a todos os nós no grupo de replicações.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&Engine=redis  
&EngineVersion=6.0  
&ReplicationGroupDescription=Demo%20cluster%20with%20replicas  
&ReplicationGroupId=sample-repl-group  
&PrimaryClusterId=redis01  
&Version=2015-02-02
```

```
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte os tópicos da APL do ElastiCache:

- [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 adicionais e parâmetros que você queira usar, consulte o tópico sobre a API do ElastiCache [CreateCacheCluster](#).

Criação de um grupo de replicação do Redis do princípio

Veja a seguir como é possível criar um grupo de replicação do Redis sem usar um cluster do Redis existente como primário. Você pode criar um grupo de replicação do Redis (modo cluster desabilitado) ou do Redis (modo cluster habilitado) a partir do princípio usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Antes de continuar, decida se deseja criar um grupo de replicação do Redis (modo cluster desabilitado) ou do Redis (modo cluster habilitado). Para obter orientação nessa decisão, consulte [Replicação: Redis \(modo cluster desabilitado\) versus Redis \(modo cluster habilitado\)](#).

Tópicos

- [Criação de um grupo de replicação do Redis \(modo cluster desabilitado\) do princípio](#)
- [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio](#)

Criação de um grupo de replicação do Redis (modo cluster desabilitado) do princípio

Você pode criar um grupo de replicação do Redis (modo cluster desabilitado) a partir do princípio usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Um grupo de replicação do Redis (modo cluster desabilitado) sempre possui um grupo de nó, um cluster primário e até cinco réplicas de leitura. Os grupos de replicação do Redis (modo cluster desabilitado) não são compatíveis com o particionamento dos seus dados.

Note

O limite de nó/fragmento pode ser aumentado para um máximo de 500 por cluster. Para solicitar um aumento de limite, consulte [Limites de serviço da AWS](#) e inclua o tipo de instância na solicitação.

Para criar um grupo de replicação do Redis (modo cluster desabilitado) do princípio, siga uma das seguintes abordagens:

Criação de um grupo de replicação do Redis (modo cluster desabilitado) do princípio (AWS CLI)

O procedimento a seguir cria um grupo de replicação do Redis (modo cluster desabilitado) usando a AWS CLI.

Ao criar um grupo de replicação do Redis (modo cluster desabilitado) do princípio, 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 do grupo de replicação do Redis (modo cluster desabilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`--replication-group-description`

Descrição do grupo de replicação.

--num-cache-clusters

O número de nós que você deseja criar com esse grupo de replicação, com réplicas primárias e de leitura combinadas.

Se você habilitar o Multi-AZ (--automatic-failover-enabled), o valor de --num-cache-clusters deverá ser pelo menos 2.

--cache-node-type

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das geração anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

--data-tiering-enabled

Defina esse parâmetro se estiver usando um tipo de nó r6gd. Se não quiser classificação de dados em níveis, defina --no-data-tiering-enabled. Para obter mais informações, consulte [Classificação de dados em níveis](#).

--cache-parameter-group

Especifique um grupo de parâmetros que corresponda à versão do seu mecanismo. Se estiver executando o Redis 3.2.4 ou posterior, especifique o grupo de parâmetros default.redis3.2 ou um grupo de parâmetros de default.redis3.2 para criar um grupo de replicação do Redis (modo cluster desabilitado). Para obter mais informações, consulte [Parâmetros específicos do Redis](#).

--network-type

ipv4, ipv6 ou dual-stack. Se você escolher pilha dupla, deverá definir o parâmetro --IpDiscovery como ipv4 ou ipv6.

--engine

redis

--engine-version

Para ter o conjunto mais amplo de recursos, escolha a versão mais recente do mecanismo.

Os nomes dos nós serão derivados do nome do grupo de replicação depois de adicionar `-00#` ao nome do grupo de replicação. Por exemplo, usando o nome do grupo de replicação `myReplGroup`, o nome para o primário será `myReplGroup-001` e as réplicas de leitura `myReplGroup-002` e `myReplGroup-006`.

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros `--transit-encryption-enabled` ou `--at-rest-encryption-enabled` e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro `--cache-subnet-group`.
- Você também deve incluir o parâmetro `--auth-token` com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

A seguinte operação cria um grupo de replicação do Redis (modo cluster desabilitado) `sample-repl-group` com três nós, um primário e duas réplicas.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --cache-node-type cache.m4.large \  
  --engine redis
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --cache-node-type cache.m4.large ^  
  --engine redis
```

A saída desse comando é semelhante a esta.

```
{
```

```
"ReplicationGroup": {
  "Status": "creating",
  "Description": "Demo cluster with replicas",
  "ClusterEnabled": false,
  "ReplicationGroupId": "sample-repl-group",
  "SnapshotRetentionLimit": 0,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "01:30-02:30",
  "MemberClusters": [
    "sample-repl-group-001",
    "sample-repl-group-002",
    "sample-repl-group-003"
  ],
  "CacheNodeType": "cache.m4.large",
  "DataTiering": "disabled",
  "PendingModifiedValues": {}
}
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico da AWS CLI [create-replication-group](#).

Criação de um grupo de replicação do Redis (modo cluster desabilitado) do princípio (API do ElastiCache)

O procedimento a seguir cria um grupo de replicação do Redis (modo cluster desabilitado) usando a API do ElastiCache.

Ao criar um grupo de replicação do Redis (modo cluster desabilitado) do princípio, você cria o grupo de replicação e todos os seus nós com uma única chamada para a operação `CreateReplicationGroup` da API do ElastiCache. Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura do grupo de replicação do Redis (modo cluster habilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.

- Não podem terminar com um hífen.

ReplicationGroupDescription

Sua descrição do grupo de replicação.

NumCacheClusters

O número total de nós que você deseja criar com esse grupo de replicação, com réplicas primárias e de leitura combinadas.

Se você habilitar o Multi-AZ (`AutomaticFailoverEnabled=true`), o valor de `NumCacheClusters` deverá ser pelo menos 2.

CacheNodeType

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das geração anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

--data-tiering-enabled

Defina esse parâmetro se estiver usando um tipo de nó `r6gd`. Se não quiser classificação de dados em níveis, defina `--no-data-tiering-enabled`. Para obter mais informações, consulte [Classificação de dados em níveis](#).

CacheParameterGroup

Especifique um grupo de parâmetros que corresponda à versão do seu mecanismo. Se estiver executando o Redis 3.2.4 ou posterior, especifique o grupo de parâmetros `default.redis3.2` ou um grupo de parâmetros de `default.redis3.2` para criar um grupo de replicação do Redis (modo cluster desabilitado). Para obter mais informações, consulte [Parâmetros específicos do Redis](#).

--network-type

`ipv4`, `ipv` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro `--IpDiscovery` como `ipv4` ou `ipv6`.

Mecanismo

`redis`

EngineVersion

6.0

Os nomes dos nós serão derivados do nome do grupo de replicação depois de adicionar `-00#` ao nome do grupo de replicação. Por exemplo, usando o nome do grupo de replicação `myReplGroup`, o nome para o primário será `myReplGroup-001` e as réplicas de leitura `myReplGroup-002` e `myReplGroup-006`.

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros `TransitEncryptionEnabled=true` ou `AtRestEncryptionEnabled=true` e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro `CacheSubnetGroup`.
- Você também deve incluir o parâmetro `AuthToken` com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

A seguinte operação cria o grupo de replicação do Redis (modo cluster desabilitado) `myReplGroup` com três nós, um primário e duas réplicas.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&CacheNodeType=cache.m4.large  
&CacheParameterGroup=default.redis6.x  
&Engine=redis  
&EngineVersion=6.0  
&NumCacheClusters=3  
&ReplicationGroupDescription=test%20group  
&ReplicationGroupId=myReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a API do ElastiCache [CreateReplicationGroup](#).

Criação de um grupo de replicação no Redis (modo cluster habilitado) do princípio

Você pode criar um cluster do Redis (modo cluster habilitado) (API/CLI: grupo de replicação) usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Um grupo de replicação do Redis (modo cluster habilitado) tem de 1 a 500 fragmentos (API/CLI: grupos de nós), um nó primário em cada fragmento e até 5 réplicas de leitura em cada fragmento. É possível criar um cluster com alto número de fragmentos e baixo número de réplicas totalizando até 90 nós por cluster. Essa configuração do cluster pode variar de 90 fragmentos e 0 réplicas para 15 fragmentos e 5 réplicas, que é o número máximo de réplicas permitidas.

O limite de nó ou fragmento pode ser aumentado para um máximo de 500 por cluster se a versão do mecanismo 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 as sub-redes no grupo de sub-redes têm um intervalo CIDR muito pequeno ou as sub-redes são compartilhadas e fortemente usadas por outros clusters. Para obter mais informações, consulte [Criação de um grupo de sub-redes](#).

Para versões abaixo de 5.0.6, o limite é 250 por cluster.

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

Criação de um cluster no Redis (modo cluster habilitado)

- [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)
- [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio \(AWS CLI\)](#)
- [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio \(API do ElastiCache\)](#)

Criação de um cluster do Redis (modo cluster habilitado) (console)

Para criar um cluster do Redis (modo cluster habilitado), consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#). Certifique-se de habilitar o modo de cluster, Cluster Mode enabled (Scale Out), e especifique pelo menos dois fragmentos e um nó de réplica em cada.

Criação de um grupo de replicação no Redis (modo cluster habilitado) do princípio (AWS CLI)

O procedimento a seguir cria um grupo de replicação do Redis (modo cluster habilitado) usando a AWS CLI.

Ao criar um grupo de replicação do Redis (modo cluster habilitado) do princípio, 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 do grupo de replicação do Redis (modo cluster habilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

`--replication-group-description`

Descrição do grupo de replicação.

`--cache-node-type`

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das gerações anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

`--data-tiering-enabled`

Defina esse parâmetro se estiver usando um tipo de nó `r6gd`. Se não quiser classificação de dados em níveis, defina `--no-data-tiering-enabled`. Para obter mais informações, consulte [Classificação de dados em níveis](#).

`--cache-parameter-group`

Especifique o grupo de parâmetros `default.redis6.x.cluster.on` ou um grupo de parâmetros derivado de `default.redis6.x.cluster.on` para criar um grupo de replicação do Redis (modo cluster habilitado). Para obter mais informações, consulte [Alterações de parâmetros do Redis 6.x](#).

`--engine`

`redis`

`--engine-version`

`3.2.4`

`--num-node-groups`

O número de grupos de nós neste grupo de replicação. Os valores válidos são 1 a 500.

 Note

O limite de nó/fragmento pode ser aumentado para um máximo de 500 por cluster. Para solicitar um aumento do limite, consulte [Limites de serviço da AWS](#) e selecione o tipo de limite "Nodes per cluster per instance type" (Nós por cluster por tipo de instância).

`--replicas-per-node-group`

O número de nós de réplica em cada grupo de nós. Os valores válidos são 0 a 5.

`--network-type`

`ipv4`, `ipv` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro `--IpDiscovery` como `ipv4` ou `ipv6`.

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros `--transit-encryption-enabled` ou `--at-rest-encryption-enabled` e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro `--cache-subnet-group`.
- Você também deve incluir o parâmetro `--auth-token` com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

A seguinte operação cria o grupo de replicação do Redis (modo cluster habilitado) `sample-repl-group` com três grupos de nós/fragmentos (`--num-node-groups`), cada um com três nós, uma réplica primária e duas réplicas de leitura (`--replicas-per-node-group`).

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-node-groups 3 \  
  --replicas-per-node-group 2 \  
  --cache-node-type cache.m4.large \  
  --engine redis \  
  --security-group-ids SECURITY_GROUP_ID \  
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --cache-node-type cache.m4.large ^  
  --engine redis ^  
  --security-group-ids SECURITY_GROUP_ID ^  
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

O comando anterior gera a seguinte saída.

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "Demo cluster with replicas",  
    "ReplicationGroupId": "sample-repl-group",  
    "SnapshotRetentionLimit": 0,  
    "AutomaticFailover": "enabled",  
    "SnapshotWindow": "05:30-06:30",  
    "MemberClusters": [  
      "sample-repl-group-0001-001",  
      "sample-repl-group-0001-002",  
      "sample-repl-group-0001-003",  
      "sample-repl-group-0002-001",  
      "sample-repl-group-0002-002",
```

```
        "sample-repl-group-0002-003",
        "sample-repl-group-0003-001",
        "sample-repl-group-0003-002",
        "sample-repl-group-0003-003"
    ],
    "PendingModifiedValues": {}
}
}
```

Quando você cria um grupo de replicação do Redis (modo cluster habilitado) do princípio, pode configurar cada fragmento do cluster usando o parâmetro `--node-group-configuration`, como mostra o exemplo a seguir, que configura dois grupos de nós (console: fragmentos). O primeiro fragmento tem dois nós: um primário e uma réplica de leitura. O segundo fragmento tem três nós: um primário e duas réplicas de leitura.

`--node-group-configuration`

A configuração para cada grupo de nós. O parâmetro `--node-group-configuration` consiste nos seguintes campos.

- `PrimaryAvailabilityZone`: a zona de disponibilidade onde o nó primário desse grupo de nós está localizado. Se esse parâmetro for omitido, o ElastiCache escolherá a zona de disponibilidade para o nó primário.

Exemplo: `us-west-2a`.

- `ReplicaAvailabilityZones`: uma lista separada por vírgulas de zonas de disponibilidade em que as réplicas de leitura estão localizadas. O número de Zonas de disponibilidade nessa lista deve corresponder ao valor de `ReplicaCount`. Se esse parâmetro for omitido, o ElastiCache escolherá as zonas de disponibilidade para os nós de réplica.

Exemplo: `"us-west-2a,us-west-2b,us-west-2c"`

- `ReplicaCount`: o número de nós de réplica nesse grupo de nós.
- `Slots`: uma string que especifica o espaço de chaves para o grupo de nós. A string está no formato `startKey-endKey`. Se esse parâmetro for omitido, o ElastiCache alocará as chaves igualmente entre os grupos de nós.

Exemplo: `"0-4999"`

A operação a seguir cria o grupo de replicação do Redis (modo cluster habilitado) `new-group` com dois grupos de nós/fragmentos (`--num-node-groups`). Ao contrário do exemplo anterior, cada grupo de nós está configurado de forma diferente do outro grupo de nós (`--node-group-configuration`).

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id new-group \  
  --replication-group-description "Sharded replication group" \  
  --engine redis \  
  --snapshot-retention-limit 8 \  
  --cache-node-type cache.m4.medium \  
  --num-node-groups 2 \  
  --node-group-configuration \  
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \  
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id new-group ^  
  --replication-group-description "Sharded replication group" ^  
  --engine redis ^  
  --snapshot-retention-limit 8 ^  
  --cache-node-type cache.m4.medium ^  
  --num-node-groups 2 ^  
  --node-group-configuration \  
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \  
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

A operação anterior gera a seguinte saída.

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "Sharded replication group",  
    "ReplicationGroupId": "rc-rg",
```

```
    "SnapshotRetentionLimit": 8,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "10:00-11:00",
    "MemberClusters": [
      "rc-rg-0001-001",
      "rc-rg-0001-002",
      "rc-rg-0002-001",
      "rc-rg-0002-002",
      "rc-rg-0002-003"
    ],
    "PendingModifiedValues": {}
  }
}
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a AWS CLI [create-replication-group](#).

Criação de um grupo de replicação no Redis (modo cluster habilitado) do princípio (API do ElastiCache)

O procedimento a seguir cria um grupo de replicação do Redis (modo cluster habilitado) usando a API do ElastiCache.

Ao criar um grupo de replicação do Redis (modo cluster habilitado) do princípio, você cria o grupo de replicação e todos os seus nós com uma única chamada para a operação `CreateReplicationGroup` da API do ElastiCache. Inclua os seguintes parâmetros.

ReplicationGroupId

O nome do grupo de replicação que você está criando.

As restrições de nomenclatura do grupo de replicação do Redis (modo cluster habilitado) são as seguintes:

- Devem conter 1 a 40 caracteres alfanuméricos ou hifens.
- Deve começar com uma letra.
- Não podem conter dois hifens consecutivos.
- Não podem terminar com um hífen.

ReplicationGroupDescription

Descrição do grupo de replicação.

NumNodeGroups

O número de grupos de nós que você deseja criar com esse grupo de replicação. Os valores válidos são 1 a 500.

ReplicasPerNodeGroup

O número de nós de réplica em cada grupo de nós. Os valores válidos são 1 a 5.

NodeGroupConfiguration

A configuração para cada grupo de nós. O parâmetro `NodeGroupConfiguration` consiste nos seguintes campos.

- `PrimaryAvailabilityZone`: a zona de disponibilidade onde o nó primário desse grupo de nós está localizado. Se esse parâmetro for omitido, o ElastiCache escolherá a zona de disponibilidade para o nó primário.

Exemplo: `us-west-2a`.

- `ReplicaAvailabilityZones`: uma lista de zonas de disponibilidade em que as réplicas de leitura estão localizadas. O número de Zonas de disponibilidade nessa lista deve corresponder ao valor de `ReplicaCount`. Se esse parâmetro for omitido, o ElastiCache escolherá as zonas de disponibilidade para os nós de réplica.
- `ReplicaCount`: o número de nós de réplica nesse grupo de nós.
- `Slots`: uma string que especifica o espaço de chaves para o grupo de nós. A string está no formato `startKey-endKey`. Se esse parâmetro for omitido, o ElastiCache alocará as chaves igualmente entre os grupos de nós.

Exemplo: `"0-4999"`

CacheNodeType

O tipo de nó para cada nó no grupo de replicação.

Os tipos de nó a seguir são compatíveis com o ElastiCache. Em linhas gerais, os tipos da geração atual fornecem mais memória e poder computacional a um custo mais baixo em comparação aos equivalentes das geração anteriores.

Para obter mais informações sobre detalhes de performance para cada tipo de nó, consulte [Tipos de instância do Amazon EC2](#).

--data-tiering-enabled

Defina esse parâmetro se estiver usando um tipo de nó r6gd. Se não quiser classificação de dados em níveis, defina `--no-data-tiering-enabled`. Para obter mais informações, consulte [Classificação de dados em níveis](#).

CacheParameterGroup

Especifique o grupo de parâmetros `default.redis6.x.cluster.on` ou um grupo de parâmetros derivado de `default.redis6.x.cluster.on` para criar um grupo de replicação do Redis (modo cluster habilitado). Para obter mais informações, consulte [Alterações de parâmetros do Redis 6.x](#).

--network-type

`ipv4`, `ipv` ou `dual-stack`. Se você escolher pilha dupla, deverá definir o parâmetro `--IpDiscovery` como `ipv4` ou `ipv6`.

Mecanismo

`redis`

EngineVersion

`6.0`

Se você quiser habilitar a criptografia em trânsito ou em repouso neste grupo de replicação, adicione um ou ambos os parâmetros `TransitEncryptionEnabled=true` ou `AtRestEncryptionEnabled=true` e cumpra as seguintes condições.

- Seu grupo de replicação deve estar executando o Redis versão 3.2.6 ou 4.0.10.
- O grupo de replicação precisa ser criado em uma Amazon VPC.
- Você também deve incluir o parâmetro `CacheSubnetGroup`.
- Você também deve incluir o parâmetro `AuthToken` com o valor de string especificado pelo cliente para o token AUTH (senha) necessário para realizar operações neste grupo de replicação.

As quebras de linha foram adicionadas para legibilidade.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup
```

```
&CacheNodeType=cache.m4.large
&CacheParameterGroup=default.redis6.xcluster.on
&Engine=redis
&EngineVersion=6.0
&NumNodeGroups=3
&ReplicasPerNodeGroup=2
&ReplicationGroupDescription=test%20group
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Para obter informações adicionais e parâmetros que você queira usar, consulte o tópico sobre a API do ElastiCache [CreateReplicationGroup](#).

Visualização dos detalhes de um grupo de replicação

Há momentos em que você pode querer visualizar os detalhes de um grupo de replicação. Você pode usar o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache. O processo do console é diferente para Redis (modo cluster desabilitado) e Redis (modo cluster habilitado).

Visualizar os detalhes de um grupo de replicação

- [Visualização de detalhes de um Redis \(modo cluster desabilitado\) com réplicas](#)
 - [Visualização de detalhes de um grupo de replicação do Redis \(modo cluster desabilitado\) \(console\)](#)
 - [Visualização dos detalhes de um grupo de replicação do Redis \(modo cluster desabilitado\) \(AWS CLI\)](#)
 - [Visualização dos detalhes de um grupo de replicação do Redis \(modo cluster desabilitado\) \(API do ElastiCache\)](#)
- [Visualização dos detalhes de um grupo de replicação do Redis \(modo cluster habilitado\)](#)
 - [Visualização de detalhes de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)
 - [Visualização de detalhes de um cluster do Redis \(modo cluster habilitado\) \(AWS CLI\)](#)
 - [Visualização de detalhes de um cluster do Redis \(modo cluster habilitado\) \(API do ElastiCache\)](#)
- [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#)
- [Visualização dos detalhes de um grupo de replicação \(API do ElastiCache\)](#)

Visualização de detalhes de um Redis (modo cluster desabilitado) com réplicas

Você pode visualizar os detalhes de um cluster do Redis (modo cluster desabilitado) com réplicas (API/CLI: grupo de replicação) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

Visualização dos detalhes de um cluster do Redis (modo cluster desabilitado)

- [Visualização de detalhes de um grupo de replicação do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Visualização dos detalhes de um grupo de replicação do Redis \(modo cluster desabilitado\) \(AWS CLI\)](#)
- [Visualização dos detalhes de um grupo de replicação do Redis \(modo cluster desabilitado\) \(API do ElastiCache\)](#)

Visualização de detalhes de um grupo de replicação do Redis (modo cluster desabilitado) (console)

Para ver os detalhes de um cluster do Redis (modo cluster desabilitado) com réplicas usando o console do ElastiCache, consulte o tópico [Visualização de detalhes de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#).

Visualização dos detalhes de um grupo de replicação do Redis (modo cluster desabilitado) (AWS CLI)

Para um exemplo de AWS CLI que exibe os detalhes de um grupo de replicação do Redis (modo cluster desabilitado), consulte [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#).

Visualização dos detalhes de um grupo de replicação do Redis (modo cluster desabilitado) (API do ElastiCache)

Para um exemplo da API do ElastiCache que exibe os detalhes de um grupo de replicação do Redis (modo cluster desabilitado), consulte [Visualização dos detalhes de um grupo de replicação \(API do ElastiCache\)](#).

Visualização dos detalhes de um grupo de replicação do Redis (modo cluster habilitado)

Visualização de detalhes de um cluster do Redis (modo cluster habilitado) (console)

Para ver os detalhes de um cluster do Redis (modo cluster habilitado) usando o console do ElastiCache, consulte [Visualização de detalhes de um cluster do Redis \(modo cluster habilitado\) \(console\)](#).

Visualização de detalhes de um cluster do Redis (modo cluster habilitado) (AWS CLI)

Para um exemplo da CLI do ElastiCache que exibe os detalhes de um grupo de replicação do Redis (modo cluster habilitado), consulte [Visualização dos detalhes de um grupo de replicação \(AWS CLI\)](#).

Visualização de detalhes de um cluster do Redis (modo cluster habilitado) (API do ElastiCache)

Para um exemplo da API do ElastiCache que exibe os detalhes de um grupo de replicação do Redis (modo cluster habilitado), consulte [Visualização dos detalhes de um grupo de replicação \(API do ElastiCache\)](#).

Visualização dos detalhes de um grupo de replicação (AWS CLI)

É possível visualizar os detalhes de um grupo de replicação usando o comando `describe-replication-groups` da AWS CLI. Use os seguintes parâmetros opcionais para refinar a listagem. A omissão dos parâmetros retorna os detalhes de até 100 grupos de replicação.

Parâmetros opcionais

- `--replication-group-id`: use este parâmetro para listar os detalhes de um grupo de replicação específico. Se o grupo de replicação especificado tiver mais de um grupo de nós, os resultados serão retornados agrupados por grupo de nós.
- `--max-items`: use este parâmetro para limitar o número de grupos de replicação listados. O valor de `--max-items` não pode ser inferior a 20 ou superior a 100.

Example

O código a seguir lista os detalhes para até 100 grupos de replicação.

```
aws elasticache describe-replication-groups
```

O código a seguir lista os detalhes para `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

O código a seguir lista os detalhes para `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

O código a seguir lista os detalhes para até 25 grupos de replicação.

```
aws elasticache describe-replication-groups --max-items 25
```

O resultado dessa operação deve ser semelhante a este (formato JSON).

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "test",
      "NodeGroups": [
        {
          "Status": "available",
          "NodeGroupMembers": [
            {
              "CurrentRole": "primary",
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
              },
              "CacheClusterId": "rg-name-001"
            },
            {
              "CurrentRole": "replica",
              "PreferredAvailabilityZone": "us-west-2b",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
              },
              "CacheClusterId": "rg-name-002"
            },
            {
              "CurrentRole": "replica",
              "PreferredAvailabilityZone": "us-west-2c",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
              },
            },
          ]
        }
      ]
    }
  ]
}
```

```

        "CacheClusterId": "rg-name-003"
      }
    ],
    "NodeGroupId": "0001",
    "PrimaryEndpoint": {
      "Port": 6379,
      "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
    }
  }
],
"ReplicationGroupId": "rg-name",
"AutomaticFailover": "enabled",
"SnapshottingClusterId": "rg-name-002",
"MemberClusters": [
  "rg-name-001",
  "rg-name-002",
  "rg-name-003"
],
"PendingModifiedValues": {}
},
{
  ... some output omitted for brevity
}
]
}

```

Para obter mais informações, consulte a AWS CLI para o tópico do ElastiCache [describe-replication-groups](#).

Visualização dos detalhes de um grupo de replicação (API do ElastiCache)

Você pode visualizar os detalhes de uma replicação usando a operação AWS CLI `DescribeReplicationGroups`. Use os seguintes parâmetros opcionais para refinar a listagem. A omissão dos parâmetros retorna os detalhes de até 100 grupos de replicação.

Parâmetros opcionais

- `ReplicationGroupId`: use este parâmetro para listar os detalhes de um grupo de replicação específico. Se o grupo de replicação especificado tiver mais de um grupo de nós, os resultados serão retornados agrupados por grupo de nós.
- `MaxRecords`: use este parâmetro para limitar o número de grupos de replicação listados. O valor de `MaxRecords` não pode ser inferior a 20 ou superior a 100. O padrão é 100.

Example

O código a seguir lista os detalhes para até 100 grupos de replicação.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

O código a seguir lista os detalhes para myReplGroup.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=myReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

O código a seguir lista os detalhes para até 25 clusters.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte o tópico de referência da API do ElastiCache [DescribeReplicationGroups](#).

Localização de endpoints de grupo de replicação

Um aplicativo pode se conectar a qualquer nó em um grupo de replicação, desde que tenha o endpoint DNS e o número da porta desse nó. Dependendo de você estar executando um grupo de replicação do Redis (modo cluster desabilitado) ou do Redis (modo cluster habilitado), você estará interessado em diferentes endpoints.

Redis (modo cluster desabilitado)

Os clusters Redis (modo cluster desabilitado) com réplicas têm três tipos de endpoints: o endpoint primário, o endpoint leitor e os endpoints de nó. O endpoint primário é um nome DNS que sempre é resolvido para o nó primário no cluster. O endpoint primário é imune às alterações no seu cluster, como promover uma réplica de leitura para a função primária. Para atividades de gravação, recomendamos que suas aplicações se conectem ao endpoint primário.

Um endpoint leitor divide uniformemente as conexões de entrada no endpoint entre todas as réplicas de leitura em um cluster do ElastiCache for Redis. Fatores adicionais, como quando o aplicativo cria as conexões ou como o aplicativo (re)usa as conexões determinarão a distribuição do tráfego. Os endpoints de leitor acompanham as alterações do cluster em tempo real à medida que réplicas são adicionadas ou removidas. Você pode colocar várias réplicas de leitura do seu cluster do ElastiCache para Redis em diferentes zonas de disponibilidade da AWS (AZ) para garantir a alta disponibilidade dos endpoints leitores.

Note

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

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

Redis (modo cluster habilitado)

Clusters do Redis (modo cluster habilitado) com réplicas, como eles têm vários fragmentos (API/CLI: grupos de nós), o que significa que também possuem vários nós primários, possuem uma estrutura de endpoint diferente dos clusters do Redis (modo cluster desabilitado). O Redis (modo cluster

habilitado) tem um endpoint de configuração que "conhece" todos os endpoints primários e de nó no cluster. Seu aplicativo conecta-se ao endpoint de configuração. Sempre que seu aplicativo grava ou lê no/do endpoint de configuração do cluster, o Redis, nos bastidores, determina a qual estilhaço a chave pertence e qual endpoint nesse estilhaço deve ser usado. O processo é bastante transparente para o seu aplicativo.

Você pode localizar os endpoints de um cluster usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Localizar endpoints de grupo de replicação

Para localizar os endpoints do seu grupo de replicação, consulte um dos seguintes tópicos:

- [Localização de endpoints de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Localização de endpoints para um cluster do Redis \(modo cluster habilitado\) \(console\)](#)
- [Encontrar os endpoints para grupos de replicação \(AWS CLI\)](#)
- [Localização de endpoints para grupos de replicação \(API do ElastiCache\)](#)

Modificação de um grupo de replicação

Restrições importantes

- Atualmente, o ElastiCache oferece suporte a modificações limitadas de um grupo de replicação do Redis (modo cluster habilitado), por exemplo, alterando a versão do mecanismo usando a operação da API `ModifyReplicationGroup` (CLI: `modify-replication-group`). Você pode modificar o número de fragmentos (grupos de nós) em um cluster do Redis (modo cluster habilitado) com a operação da API [ModifyReplicationGroupShardConfiguration](#) (CLI: `modify-replication-group-shard-configuration`). Para obter mais informações, consulte [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#).

Outras modificações em um cluster do Redis (modo cluster habilitado) exigem que você o crie novamente com o novo cluster que incorpore as alterações.

- Você pode atualizar os clusters e grupos de replicação do Redis (modo cluster desabilitado) e Redis (modo cluster habilitado) para versões mais recentes do mecanismo. No entanto, você não pode fazer downgrade para versões anteriores do mecanismo, exceto excluindo o cluster existente ou grupo de replicação e criando-o novamente. Para obter mais informações, consulte [Versões de mecanismos e atualização](#).
- Você pode atualizar um cluster existente do ElastiCache para Redis que usa o modo de cluster desativado para usar o modo cluster ativado, usando o console, a API [ModifyReplicationGroup](#) ou o comando da CLI `modify-replication-group`, conforme mostrado no exemplo abaixo. Ou você pode seguir as etapas em [Modificar o modo de cluster](#).

Você pode modificar um cluster do Redis (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache. Atualmente, o ElastiCache oferece suporte para um número limitado de modificações em um grupo de replicação do Redis (modo cluster habilitado). Outras modificações exigem que você crie um backup do grupo de replicação e, em seguida, use esse backup para propagar um novo grupo de replicação do Redis (modo cluster habilitado).

Tópicos

- [Usar a AWS Management Console](#)

- [Usar a AWS CLI](#)
- [Uso da API do ElastiCache](#)

Usar a AWS Management Console

Para modificar um cluster do Redis (modo cluster desabilitado), consulte [Modificando um cluster ElastiCache](#).

Usar a AWS CLI

Veja a seguir exemplos de AWS CLI do comando `modify-replication-group`. Você pode usar o mesmo comando para fazer outras modificações em um grupo de replicação.

Ativar multi-AZ em um grupo de replicação do Redis existente:

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --multi-az-enabled = true
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --multi-az-enabled
```

Modifique o modo de cluster de desativado para ativado:

Para modificar o modo de cluster de desativado para ativado, você deve primeiro definir o modo de cluster como compatível. O modo compatível permite que seus clientes Redis se conectem usando o modo de cluster ativado e o modo de cluster desativado. Depois de migrar todos os clientes Redis para usar o modo de cluster ativado, você poderá 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 \  
  --cluster-mode = compatible
```

```
--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 comando `modify-replication-group` da AWS CLI, consulte [modify-replication-group](#) ou [Modificar o modo de cluster](#) no Guia do usuário do ElastiCache para Redis.

Uso da API do ElastiCache

A seguinte operação da API do ElastiCache habilita o recurso multi-AZ em um grupo de replicação do Redis 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  
&Multi-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 operação da API do ElastiCache `ModifyReplicationGroup`, consulte [ModifyReplicationGroup](#).

Exclusão de um grupo de replicação

Se você não precisar mais de um dos seus clusters com réplicas (chamados de grupos de replicação na API/CLI), poderá excluí-lo. Quando você exclui um grupo de replicação, o ElastiCache exclui todos os nós nesse grupo.

Depois de ter iniciado essa operação, ela não poderá ser interrompida ou cancelada.

Warning

Ao excluir um cluster do ElastiCache for Redis, seus snapshots manuais são retidos. Você também poderá criar um snapshot final antes da exclusão do cluster. Os snapshots de cache automáticos não são retidos.

Exclusão de um grupo de replicação (Console)

Para excluir um cluster que possui réplicas, consulte [Excluir um cluster](#).

Excluir um grupo de replicação (AWS CLI)

Use o comando [delete-replication-group](#) para excluir um grupo de replicação.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Um prompt solicita que você confirme sua decisão. Insira y (sim) para iniciar a operação imediatamente. Após o início do processo, ele é irreversível.

```
After you begin deleting this replication group, all of its nodes will be deleted as well.
```

```
Are you sure you want to delete this replication group? [Ny]y
```

```
REPLICATIONGROUP my-repgroup My replication group deleting
```

Exclusão de um grupo de replicação (API do ElastiCache)

Chame [DeleteReplicationGroup](#) com o parâmetro `ReplicationGroup`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DeleteReplicationGroup
&ReplicationGroupId=my-repgroup
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Note

Se você definir o parâmetro `RetainPrimaryCluster` como `true`, todas as réplicas de leitura serão excluídas, mas o cluster primário será mantido.

Alteração do número de réplicas

Você pode aumentar ou diminuir dinamicamente o número de réplicas de leitura no grupo de replicação Redis usando o AWS Management Console, a AWS CLI ou a API do ElastiCache. Caso seu grupo de replicação seja um Redis (modo cluster habilitado), você pode escolher de quais fragmentos (grupos de modo) aumentar ou diminuir o número de réplicas.

Para alterar dinamicamente o número de réplicas no grupo de replicação Redis, escolha a operação na tabela a seguir que se adapte à situação.

Para fazer isso	Para o Redis (modo cluster habilitado)	Para o Redis (modo cluster desabilitado)
Adicionar réplicas	Aumento do número de réplicas em um fragmento	Aumento do número de réplicas em um fragmento Adicionando uma réplica de leitura, para grupos de replicação do Redis (Modo cluster desabilitado)
Excluir réplicas	Diminuição do número de réplicas em um fragmento	Diminuição do número de réplicas em um fragmento Exclusão de uma réplica de leitura, para grupos de replicação do Redis (modo cluster desabilitado)

Aumento do número de réplicas em um fragmento

Você pode aumentar o número de réplicas em um fragmento do Redis (modo cluster habilitado) ou um grupo de replicação do Redis (modo cluster desabilitado) até um máximo de cinco. Você pode fazer isso usando o AWS Management Console, a AWS CLI ou a API do ElastiCache.

Tópicos

- [Usar a AWS Management Console](#)
- [Usar a AWS CLI](#)
- [Uso da API do ElastiCache](#)

Usar a AWS Management Console

O procedimento a seguir usa o console para aumentar o número de réplicas em um grupo de replicação do Redis (modo cluster habilitado).

Para aumentar o número de réplicas em estilhaços do Redis

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis e o nome do grupo de replicação ao qual você deseja adicionar réplicas.
3. Escolha a caixa de cada estilhaço a que você deseja adicionar réplicas.
4. Escolha Add replicas (Adicionar réplicas).
5. Conclua a página Add Replicas to Shards (Adicionar réplicas a estilhaços):
 - Em New number of replicas/shard (Novo número de réplicas/estilhaço), informe o número de réplicas que todos os estilhaços selecionados devem ter. Esse valor deve ser maior que ou igual a Current Number of Replicas per shard (Número atual de réplicas por estilhaço) e menor que ou igual a cinco. Recomendamos pelo menos duas réplicas para trabalho mínimo.
 - Em Availability Zones (Zonas de disponibilidade), escolha No preference (Sem preferência) para o ElastiCache escolher uma zona de disponibilidade para cada nova réplica, ou Specify Availability Zones (Especificar zonas de disponibilidade) a fim de escolher uma zona de disponibilidade para cada nova réplica.

Caso você escolha Specify Availability Zones (Especificar zonas de disponibilidade), para cada nova réplica, especifique uma zona de disponibilidade usando a lista.

6. Escolha Add (Adicionar) a fim de adicionar as réplicas ou Cancel (Cancelar) a fim de cancelar a operação.

Usar a AWS CLI

Para aumentar o número de réplicas em um estilhaço do Redis, use o comando `increase-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: obrigatório. Identifica em qual grupo de replicação você deseja aumentar o número de réplicas.
- `--apply-immediately` ou `--no-apply-immediately`: obrigatório. Especifica se é necessário aumentar a contagem de réplicas imediatamente (`--apply-immediately`) ou na próxima janela de manutenção (`--no-apply-immediately`). Atualmente, `--no-apply-immediately` não é suportado.
- `--new-replica-count`: optional. Especifica o número de nós de réplica que você deseja concluir, até um máximo de cinco. Use este parâmetro em grupos de replicação do Redis (modo cluster desabilitado) nos quais haja apenas um grupo de nós ou o grupo do Redis (modo cluster habilitado), ou onde você deseja que todos os grupos de nós tenham o mesmo número de réplicas. Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- `--replica-configuration`: optional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use este parâmetro em grupos do Redis (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

`--replica-configuration` tem três membros opcionais:

- `NodeId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Redis (modo cluster desabilitado), o ID de fragmento é sempre `0001`. Para encontrar o ID do grupo de nós do Redis (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- `NewReplicaCount`: o número de réplicas que você deseja nesse grupo de nós ao final desta operação. O valor deve ser maior que o número atual de réplicas, até um máximo de cinco. Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- `PreferredAvailabilityZones`: uma lista de strings de `PreferredAvailabilityZone` que especificam em quais zonas de disponibilidade os nós do grupo de replicação devem

estar. O número de valores `PreferredAvailabilityZone` deve ser igual ao valor de `NewReplicaCount` mais 1 para contabilizar o nó primário. Caso esse membro de `--replica-configuration` seja omitido, o ElastiCache for Redis escolherá a zona de disponibilidade para cada uma das novas réplicas.

 Important

Você deve incluir o parâmetro `--new-replica-count` ou `--replica-configuration`, mas não ambos, na chamada.

Example

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para três. Quando o exemplo é concluído, existem três réplicas em cada grupo de nós. Esse número se aplica caso este seja um grupo do Redis (modo cluster desabilitado) com um grupo de nós único ou um grupo do Redis (modo cluster habilitado) com vários grupos de nós.

Para Linux, macOS ou Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id sample-repl-group \  
  --new-replica-count 3 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --new-replica-count 3 ^  
  --apply-immediately
```

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Redis (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

Para Linux, macOS ou Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id sample-repl-group \  
  --replica-configuration \  
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \  
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \  
  --apply-immediately
```

Para Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --replica-configuration ^  
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^  
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \  
  --apply-immediately
```

Para obter mais informações sobre como aumentar o número de réplicas usando a CLI, consulte [increase-replica-count](#) na Referência da linha de comando do Amazon ElastiCache.

Uso da API do ElastiCache

Para aumentar o número de réplicas em um estilhaço do Redis, use a ação `IncreaseReplicaCount` com os seguintes parâmetros:

- `ReplicationGroupId`: obrigatório. Identifica em qual grupo de replicação você deseja aumentar o número de réplicas.
- `ApplyImmediately`: obrigatório. Especifica se é necessário aumentar a contagem de réplicas imediatamente (`ApplyImmediately=True`) ou na próxima janela de manutenção (`ApplyImmediately=False`). Atualmente, `ApplyImmediately=False` não é suportado.
- `NewReplicaCount`: optional. Especifica o número de nós de réplica que você deseja concluir, até um máximo de cinco. Use este parâmetro em grupos de replicação do Redis (modo cluster desabilitado) nos quais haja apenas um grupo de nós ou nos grupos do Redis (modo cluster habilitado) onde você deseja que todos os grupos de nós tenham o mesmo número de réplicas.

Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.

- `ReplicaConfiguration`: optional. Permite definir o número de réplicas e as zonas de disponibilidade de cada grupo de nós independentemente. Use este parâmetro em grupos do Redis (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

`ReplicaConfiguraion` tem três membros opcionais:

- `NodeId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Redis (modo cluster desabilitado), o ID do grupo de nós (fragmento) é sempre `0001`. Para encontrar o ID do grupo de nós do Redis (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- `NewReplicaCount`: o número de réplicas que você deseja nesse grupo de nós ao final desta operação. O valor deve ser maior que o número atual de réplicas e um máximo de cinco. Caso esse valor não seja maior que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- `PreferredAvailabilityZones`: uma lista de strings de `PreferredAvailabilityZone` que especificam em quais zonas de disponibilidade os nós do grupo de replicação devem estar. O número de valores `PreferredAvailabilityZone` deve ser igual ao valor de `NewReplicaCount` mais 1 para contabilizar o nó primário. Caso esse membro de `ReplicaConfiguration` seja omitido, o ElastiCache for Redis escolherá a zona de disponibilidade para cada uma das novas réplicas.

Important

Você deve incluir o parâmetro `NewReplicaCount` ou `ReplicaConfiguration`, mas não ambos, na chamada.

Example

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para três. Quando o exemplo é concluído, existem três réplicas em cada grupo de nós. Esse número se aplica caso este seja um grupo do Redis (modo cluster desabilitado) com um grupo de nós único ou um grupo do Redis (modo cluster habilitado) com vários grupos de nós.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=True  
  &NewReplicaCount=3  
  &ReplicationGroupId=sample-repl-group  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

O exemplo a seguir aumenta o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Redis (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=IncreaseReplicaCount  
  &ApplyImmediately=True  
  &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001  
  &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=  
east-1a  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=  
east-1c  
  
  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=  
east-1b  
  &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003  
  &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3  
  
  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=  
east-1a  
  
  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=  
east-1b
```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=  
east-1c
```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=  
east-1c
```

```
  &ReplicationGroupId=sample-repl-group
```

```
  &Version=2015-02-02
```

```
  &SignatureVersion=4
```

```
  &SignatureMethod=HmacSHA256
```

```
  &Timestamp=20150202T192317Z
```

```
  &X-Amz-Credential=<credential>
```

Para obter mais informações sobre como aumentar o número de réplicas usando a API, consulte [IncreaseReplicaCount](#) na Referência da API do Amazon ElastiCache.

Diminuição do número de réplicas em um fragmento

Você pode reduzir o número de réplicas em um fragmento do Redis (modo cluster habilitado) ou um grupo de replicação do Redis (modo cluster desabilitado):

- Para o Redis (modo cluster desabilitado), é possível reduzir o número de réplicas para um, caso o multi-AZ esteja habilitado, e para zero, caso ele não esteja habilitado.
- Para o Redis (modo cluster habilitado), você pode diminuir o número de réplicas a zero. No entanto, você não pode fazer failover para uma réplica, caso o nó primário falhe.

Você pode usar o AWS Management Console, a AWS CLI ou a API do ElastiCache para reduzir o número de réplicas em um grupo de nós (fragmento) ou em um grupo de replicação.

Tópicos

- [Usar a AWS Management Console](#)
- [Usar a AWS CLI](#)
- [Uso da API do ElastiCache](#)

Usar a AWS Management Console

O procedimento a seguir usa o console para reduzir o número de réplicas em um grupo de replicação do Redis (modo cluster habilitado).

Para diminuir o número de réplicas em um estilhaço do Redis

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis e o nome do grupo de replicação do qual você deseja excluir réplicas.
3. Escolha a caixa de cada estilhaço de que você deseja excluir um nó de réplica.
4. Escolha Delete replicas (Excluir réplicas).
5. Conclua a página Delete Replicas from Shards (Excluir réplicas de estilhaços):
 - a. Em New number of replicas/shard (Novo número de réplicas/estilhaço), informe o número de réplicas que os estilhaços selecionados devem ter. Esse número deve ser maior que ou igual a 1. Recomendamos pelo menos duas réplicas por estilhaço para trabalho mínimo.

- b. Escolha Delete (Excluir) a fim de excluir as réplicas ou Cancel (Cancelar) a fim de cancelar a operação.

Important

- Caso você não especifique os nós de réplica a serem excluídos, o ElastiCache for Redis selecionará automaticamente os nós de réplica para exclusão. Ao fazer isso, o ElastiCache for Redis tenta manter a arquitetura multi-AZ do grupo de replicação seguido preservando réplicas com atraso de replicação mínimo em relação ao principal.
- Não é possível excluir o primário ou os nós primários em um grupo de replicação. Se você especificar o nó primário para exclusão, a operação falhará com um evento de erro indicando que o nó primário foi selecionado para exclusão.

Usar a AWS CLI

Para diminuir o número de réplicas em um estilhaço do Redis, use o comando `decrease-replica-count` com os seguintes parâmetros:

- `--replication-group-id`: obrigatório. Identifica em qual grupo de replicação você deseja diminuir o número de réplicas.
- `--apply-immediately` ou `--no-apply-immediately`: obrigatório. Especifica se é necessário diminuir a contagem de réplicas imediatamente (`--apply-immediately`) ou na próxima janela de manutenção (`--no-apply-immediately`). Atualmente, `--no-apply-immediately` não é suportado.
- `--new-replica-count`: 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 IDs de nó que especificam 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 este parâmetro em grupos do Redis (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

`--replica-configuration` tem três membros opcionais:

- `NodeId`: o ID de quatro dígitos do grupo de nós que você está configurando. Para grupos de replicação do Redis (modo cluster desabilitado), o ID de fragmento é sempre `0001`. Para encontrar o ID do grupo de nós do Redis (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- `NewReplicaCount`: um parâmetro opcional que especifica o número de nós de réplica desejado. O valor de `NewReplicaCount` deve ser um valor válido menos o número atual de réplicas nos grupos de nós. Para valores permitidos mínimos, consulte [Diminuição do número de réplicas em um fragmento](#). Caso o valor de `NewReplicaCount` não atenda a esse requisito, a chama falha.
- `PreferredAvailabilityZones`: uma lista de strings de `PreferredAvailabilityZone` que especificam em quais zonas de disponibilidade os nós do grupo de replicação estão. O número de valores `PreferredAvailabilityZone` deve ser igual ao valor de `NewReplicaCount` mais 1 para contabilizar o nó primário. Caso esse membro de `--replica-configuration` seja omitido, o ElastiCache for Redis escolherá a zona de disponibilidade para cada uma das novas réplicas.

Important

Você deve incluir um, e somente um, parâmetro `--new-replica-count`, `--replicas-to-remove`, ou `--replica-configuration`.

Example

O exemplo a seguir usa `--new-replica-count` para diminuir o número de réplicas no grupo de replicação `sample-repl-group` para um. Quando o exemplo é concluído, existe uma réplica em cada grupo de nós. Esse número se aplica caso este seja um grupo do Redis (modo cluster desabilitado) com um grupo de nós único ou um grupo do Redis (modo cluster habilitado) com vários grupos de nós.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count
  --replication-group-id sample-repl-group \
  --new-replica-count 1 \
```

```
--apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --new-replica-count 1 ^
  --apply-immediately
```

O exemplo a seguir diminui o número de réplicas no grupo de replicação `sample-repl-group` removendo duas réplicas especificadas (`0001` e `0003`) do grupo de nós.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id sample-repl-group \
  --replicas-to-remove 0001,0003 \
  --apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --replicas-to-remove 0001,0003 \
  --apply-immediately
```

O exemplo a seguir usa `--replica-configuration` a fim de diminuir o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Redis (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

Para Linux, macOS ou Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id sample-repl-group \
  --replica-configuration \
    NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-  
east-1c \
```

```
NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \  
--apply-immediately
```

Para Windows:

```
aws elasticache decrease-replica-count ^  
--replication-group-id sample-repl-group ^  
--replica-configuration ^  
NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c ^  
NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \  
--apply-immediately
```

Para obter mais informações sobre como reduzir o número de réplicas usando a CLI, consulte [decrease-replica-count](#) na Referência da linha de comando do Amazon ElastiCache.

Uso da API do ElastiCache

Para diminuir o número de réplicas em um estilhaço do Redis, use a ação `DecreaseReplicaCount` com os seguintes parâmetros:

- `ReplicationGroupId`: obrigatório. Identifica em qual grupo de replicação você deseja diminuir o número de réplicas.
- `ApplyImmediately`: obrigatório. Especifica se é necessário diminuir a contagem de réplicas imediatamente (`ApplyImmediately=True`) ou na próxima janela de manutenção (`ApplyImmediately=False`). Atualmente, `ApplyImmediately=False` não é suportado.
- `NewReplicaCount`: 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 IDs de nó que especificam 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 este parâmetro em grupos do Redis (modo cluster habilitado) onde você deseja configurar cada grupo de nós independentemente.

ReplicaConfiguration tem três membros opcionais:

- **NodeId:** o ID de quatro dígitos do grupo de nós que você está configurando. Para o grupos de replicação do Redis (modo cluster desabilitado), o ID do grupo de nós é sempre 0001. Para encontrar o ID do grupo de nós do Redis (modo cluster habilitado), consulte [Localização de um ID de fragmento](#).
- **NewReplicaCount:** o número de réplicas que você deseja nesse grupo de nós ao final desta operação. O valor deve ser menor que o número atual de réplicas até um mínimo de 1, caso o Multi-AZ esteja habilitado ou 0 caso o Multi-AZ com failover automático não esteja habilitado. Caso esse valor não seja menor que o número atual de réplicas no grupo de nós, a chamada falha com uma exceção.
- **PreferredAvailabilityZones:** uma lista de strings de PreferredAvailabilityZone que especificam em quais zonas de disponibilidade os nós do grupo de replicação estão. O número de valores PreferredAvailabilityZone deve ser igual ao valor de NewReplicaCount mais 1 para contabilizar o nó primário. Caso esse membro de ReplicaConfiguration seja omitido, o ElastiCache for Redis escolherá a zona de disponibilidade para cada uma das novas réplicas.

Important

Você deve incluir um, e somente um, parâmetro NewReplicaCount, ReplicasToRemove, ou ReplicaConfiguration.

Example

O exemplo a seguir usa NewReplicaCount para diminuir o número de réplicas no grupo de replicação `sample-repl-group` para um. Quando o exemplo é concluído, existe uma réplica em cada grupo de nós. Esse número se aplica caso este seja um grupo do Redis (modo cluster desabilitado) com um grupo de nós único ou um grupo do Redis (modo cluster habilitado) com vários grupos de nós.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DecreaseReplicaCount  
  &ApplyImmediately=True  
  &NewReplicaCount=1  
  &ReplicationGroupId=sample-repl-group
```

```
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

O exemplo a seguir diminui o número de réplicas no grupo de replicação `sample-repl-group` removendo duas réplicas especificadas (`0001` e `0003`) do grupo de nós.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&ReplicasToRemove.ReplicaToRemove.1=0001
&ReplicasToRemove.ReplicaToRemove.2=0003
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

O exemplo a seguir usa `ReplicaConfiguration` a fim de diminuir o número de réplicas no grupo de replicação `sample-repl-group` para o valor especificado para os dois grupos de nós especificados. Como há vários grupos de nós, este é um grupo de replicação do Redis (modo cluster habilitado). Durante a especificação do `PreferredAvailabilityZones` opcional, o número de zonas de disponibilidade listadas deve ser igual ao valor de `NewReplicaCount` mais 1. Essa abordagem contabiliza o nó primário do grupo identificado por `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DecreaseReplicaCount
&ApplyImmediately=True
&ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
&ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
&ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
&ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2
```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
  &ReplicationGroupId=sample-repl-group
  &Version=2015-02-02
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20150202T192317Z
  &X-Amz-Credential=<credential>
```

Para obter mais informações sobre como reduzir o número de réplicas usando a API, consulte [DecreaseReplicaCount](#) na Referência da API do Amazon ElastiCache.

Adicionando uma réplica de leitura, para grupos de replicação do Redis (Modo cluster desabilitado)

As informações no tópico a seguir só se aplicam a grupos de replicação do Redis (modo cluster desabilitado).

À medida que seu tráfego de leitura aumentar, talvez você queira espalhar essas leituras em mais nós e reduzir a pressão de leitura em qualquer nó. Neste tópico, você pode descobrir como adicionar uma réplica de leitura a um cluster do Redis (modo cluster desabilitado).

Um grupo de replicação do Redis (modo cluster desabilitado) pode ter no máximo cinco réplicas de leitura. Se você tentar adicionar uma réplica de leitura a um grupo de replicação que já tenha cinco réplicas de leitura, a operação falhará.

Para obter informações sobre como adicionar réplicas a um grupo de replicação do Redis (modo cluster habilitado), consulte o seguinte:

- [Escalabilidade de culsters no Redis \(modo cluster habilitado\)](#)
- [Aumento do número de réplicas em um fragmento](#)

Você pode adicionar uma réplica de leitura a um cluster do Redis (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Tópicos relacionados

- [Adição de nós a um cluster](#)
- [Adição de uma réplica de leitura a um grupo de replicação \(AWS CLI\)](#)
- [Adição de uma réplica de leitura a um grupo de replicação usando a API](#)

Adição de uma réplica de leitura a um grupo de replicação (AWS CLI)

Para adicionar uma réplica de leitura a um grupo de replicação do Redis (modo cluster desabilitado), use o comando da AWS CLI `create-cache-cluster` com o parâmetro `--replication-group-id` para especificar a qual grupo de replicação adicionar o cluster (nó).

O exemplo a seguir cria o cluster `my-read-replica` e o adiciona ao grupo de replicação `my-replication-group`. Os tipos de nós, os grupos de parâmetros, os grupos de segurança, as janelas de manutenção e outras configurações da réplica de leitura são os mesmos dos outros nós em `my-replication-group`.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-read-replica \  
  --replication-group-id my-replication-group
```

Para Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-read-replica ^  
  --replication-group-id my-replication-group
```

Para obter mais informações sobre como adicionar uma réplica de leitura usando a CLI, consulte [create-cache-cluster](#) na Referência da linha de comando do Amazon ElastiCache.

Adição de uma réplica de leitura a um grupo de replicação usando a API

Para adicionar uma réplica de leitura a um grupo de replicação do Redis (modo cluster desabilitado), use a operação `CreateCacheCluster` do ElastiCache com o parâmetro `ReplicationGroupId` para especificar a qual grupo de replicação adicionar o cluster (nó).

O exemplo a seguir cria o cluster `myReadReplica` e o adiciona ao grupo de replicação `myReplicationGroup`. Os tipos de nós, os grupos de parâmetros, os grupos de segurança, as

janelas de manutenção e outras configurações da réplica de leitura são os mesmos dos outros nós `myReplicationGroup`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=myReadReplica  
&ReplicationGroupId=myReplicationGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações sobre como adicionar uma réplica de leitura usando a API, consulte [CreateCacheCluster](#) na Referência da API do Amazon ElastiCache.

Exclusão de uma réplica de leitura, para grupos de replicação do Redis (modo cluster desabilitado)

As informações no tópico a seguir só se aplicam a grupos de replicação do Redis (modo cluster desabilitado).

À medida que o tráfego de leitura no seu grupo de replicação Redis mudar, talvez você queira adicionar ou remover réplicas de leitura. A remoção de um nó de um grupo de replicação do Redis (modo cluster desabilitado) é o mesmo que simplesmente excluir um cluster, embora existam restrições:

- Não é possível remover o primário de um grupo de replicação. Se quiser excluir o primário, você deverá fazer o seguinte:
 1. Promover uma réplica de leitura para primário. Para obter mais informações sobre como promover uma réplica de leitura para primário, consulte [Promoção de uma réplica de leitura a primário, para grupos de replicação do Redis \(modo cluster desabilitado\)](#).
 2. Exclua o primário antigo. Para uma restrição nesse método, consulte o próximo ponto.
- Se o recurso Multi-AZ estiver habilitado em um grupo de replicação, você não poderá remover a última réplica de leitura do grupo de replicação. Nesse caso, faça o seguinte:
 1. Modifique o grupo de replicação desabilitando o Multi-AZ. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
 2. Exclua a réplica de leitura.

Você pode remover uma réplica de leitura de um grupo de replicação do Redis (modo cluster desabilitado) usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

Para obter orientações sobre como excluir um cluster de um grupo de replicação Redis, consulte o seguinte:

- [Usar a AWS Management Console](#)
- [Usar a AWS CLI](#)
- [Uso da API do ElastiCache](#)
- [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#)
- [Diminuição do número de réplicas em um fragmento](#)

Promoção de uma réplica de leitura a primário, para grupos de replicação do Redis (modo cluster desabilitado)

As informações no tópico a seguir só se aplicam a grupos de replicação do Redis (modo cluster desabilitado).

Você pode promover uma réplica de leitura de um cluster do Redis (modo cluster desabilitado) a primário usando o AWS Management Console, a AWS CLI ou a API do ElastiCache. Você não pode promover uma réplica de leitura para primária enquanto o multi-AZ com failover automático estiver habilitado no grupo de replicação do Redis (modo cluster desabilitado). Para promover uma réplica do Redis (modo cluster desabilitado) para primário em um grupo de replicação habilitado para multi-AZ, faça o seguinte:

1. Modifique o grupo de replicação para desabilitar Multi-AZ (isso não requer que todos os clusters estejam na mesma zona de disponibilidade). Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
2. Promova a réplica de leitura para primário.
3. Modifique o grupo de replicação para reabilitar o Multi-AZ.

O Multi-AZ não está disponível em grupos de replicação que executam o Redis 2.6.13 ou anterior.

Usar a 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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Se a réplica que você quiser promover for membro de um grupo de replicação do Redis (modo cluster desabilitado) em que o multi-AZ estiver habilitado, modifique o grupo de replicação para desabilitar o multi-AZ para continuar. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).
3. Escolha Redis, na lista de clusters, escolha o grupo de replicação que você deseja modificar. Esse grupo de replicação deve estar executando o mecanismo "Redis", e não o mecanismo "Clustered Redis", e deve ter dois ou mais nós.
4. Na lista de nós, escolha o nó de réplica que você deseja promover para primária, em Actions (Ações), escolha Promote (Promover).

5. Na caixa de diálogo Promote Read Replica (Promover réplica de leitura), faça o seguinte:
 - a. Em Apply Immediately (Aplicar imediatamente), escolha Yes (Sim) para promover a réplica de leitura imediatamente, ou No (Não) para promovê-la na próxima janela de manutenção do cluster.
 - b. Escolha Promote para promover a réplica de leitura ou Cancel para cancelar a operação.
6. Caso o cluster tenha Multi-AZ habilitado antes de você começar o processo de promoção, aguarde o status do grupo de replicação estar available (disponível), modifique o cluster para reabilitar Multi-AZ. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

Usar a AWS CLI

Você não pode promover uma réplica de leitura para primária caso o grupo de replicação esteja habilitado para Multi-AZ. Em alguns casos, a réplica que você deseja promover pode ser um membro de um grupo de replicação onde o recurso do Multi-AZ está habilitado. Nesses casos, você deve modificar o grupo de replicação para desabilitar o recurso do Multi-AZ antes de continuar. Isso não requer que todos os clusters estejam na mesma zona de disponibilidade. Para obter mais informações sobre como modificar um grupo de replicação, consulte [Modificação de um grupo de replicação](#).

O seguinte comando AWS CLI modifica o grupo de replicação `sample-repl-group`, tornando a réplica de leitura `my-replica-1` o primário no grupo de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id sample-repl-group \  
  --primary-cluster-id my-replica-1
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --primary-cluster-id my-replica-1
```

Para obter mais informações sobre como modificar um grupo de replicação, consulte [modify-replication-group](#) na Referência da linha de comando do Amazon ElastiCache.

Uso da API do ElastiCache

Você não pode promover uma réplica de leitura para primária caso o grupo de replicação esteja habilitado para Multi-AZ. Em alguns casos, a réplica que você deseja promover pode ser um membro de um grupo de replicação onde o recurso do Multi-AZ está habilitado. Nesses casos, você deve modificar o grupo de replicação para desabilitar o recurso do Multi-AZ antes de continuar. Isso não requer que todos os clusters estejam na mesma zona de disponibilidade. Para obter mais informações sobre como modificar um grupo de replicação, consulte [Modificação de um grupo de replicação](#).

A seguinte ação da API do ElastiCache modifica o grupo de replicação myRep1Group, tornando a réplica de leitura myReplica-1 o primário 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 como modificar um grupo de replicação, consulte [ModifyReplicationGroup](#) na Referência da API do Amazon ElastiCache.

Gerenciamento da manutenção

Cada cluster e grupo de replicação tem uma janela de manutenção semanal durante a qual todas as alterações do sistema são aplicadas. Se você não especificar uma janela de manutenção preferencial ao criar ou modificar um cluster ou grupo de replicação, o ElastiCache atribuirá uma janela de manutenção de 60 minutos dentro da janela de manutenção de sua região em um dia da semana escolhido aleatoriamente.

A janela de manutenção de 60 minutos é escolhida aleatoriamente entre um período de 8 horas por região. A tabela a seguir lista os blocos de tempo de cada região dos quais as janelas de

manutenção padrão são atribuídas. Você pode escolher uma janela de manutenção preferida fora do bloco de janelas de manutenção da região.

Código da região	Nome da região	Janela de manutenção da região
ap-northeast-1	Região Ásia-Pacífico (Tóquio)	13h às 21h (UTC)
ap-northeast-2	Região Ásia-Pacífico (Seul)	12h às 20h (UTC)
ap-northeast-3	Região Ásia-Pacífico (Osaka)	12h às 20h (UTC)
ap-southeast-3	Região Ásia-Pacífico (Jacarta)	De 14:00 a 22:00 (UTC)
ap-south-1	Região Ásia-Pacífico (Mumbai)	17h30 à 1h30 UTC
ap-southeast-1	Região Ásia-Pacífico (Singapura)	De 14:00 a 22:00 (UTC)
cn-north-1	Região China (Pequim)	14h às 22h (UTC)
cn-northwest-1	Região China (Ningxia)	De 14:00 a 22:00 (UTC)
ap-east-1	Região Ásia-Pacífico (Hong Kong)	De 13h às 21h (UTC)
ap-southeast-2	Região Ásia-Pacífico (Sydney)	12h às 20h (UTC)
eu-west-3	Região Europa (Paris)	De 23:59 a 07:29 UTC
af-south-1	Região África (Cidade do Cabo)	De 13h às 21h (UTC)
eu-central-1	Região Europa (Frankfurt)	23h às 7h (UTC)
eu-west-1	Região Europa (Irlanda)	22h às 6h (UTC)
eu-west-2	Região Europa (Londres)	23h às 7h (UTC)
me-south-1	Região Oriente Médio (Bahrein)	De 13h às 21h (UTC)
me-central-1	Região do Oriente Médio (Emirados Árabes Unidos)	De 13h às 21h (UTC)
eu-south-1	Região Europa (Milão)	De 21:00 a 05:00 (UTC)

Código da região	Nome da região	Janela de manutenção da região
sa-east-1	Região América do Sul (São Paulo)	1h às 9h UTC
us-east-1	Região Leste dos EUA (N. da Virgínia)	3h às 7h (UTC)
us-east-2	Região Leste dos EUA (Ohio)	5h às 12h UTC
us-gov-west-1	Região AWS GovCloud (US)	6h às 14h (UTC)
us-west-1	Região Oeste dos EUA (Norte da Califórnia).	6h às 14h (UTC)
us-west-2	Região Oeste dos EUA (Oregon)	6h às 14h (UTC)

Como alterar a janela de manutenção do cluster ou grupo de replicação

A janela de manutenção deve ser definida no horário de menor utilização e, portanto, talvez precise ser modificada de vez em quando. Você pode modificar o cluster ou grupo de replicação para especificar um intervalo de tempo de até 24 horas de duração durante o qual todas as atividades de manutenção solicitadas devem ocorrer. Todas as modificações de cluster diferidas ou pendentes que você tiver solicitado ocorrem durante esse período.

Note

Se você quiser aplicar modificações de tipo de nó e/ou atualizações de mecanismo imediatamente usando o AWS Management Console, selecione a caixa Apply immediately (Aplicar imediatamente). Do contrário, essas modificações serão aplicadas durante o próximo período de manutenção programada. Para usar a API, consulte [modify-replication-group](#) ou [modify-cache-cluster](#).

Mais informações

Para obter informações sobre sua janela de manutenção e substituição de nó, consulte:

- [Manutenção do ElastiCache](#): perguntas frequentes sobre manutenção e substituição de nó

- [Substituição de nós](#): Gerenciamento de substituição de nó
- [Modificação de um grupo de replicação](#): Alteração da janela de manutenção de um grupo de replicação

Configuração de parâmetros do mecanismo usando grupos de parâmetros

O Amazon ElastiCache usa parâmetros para controlar as propriedades de tempo de execução de seus nós e clusters. Geralmente, as versões mais recentes do mecanismo incluem parâmetros adicionais para dar suporte à funcionalidade mais recente. Para tabelas de parâmetros, consulte [Parâmetros específicos do Redis](#).

Como seria de se esperar, alguns valores de parâmetros, como `maxmemory`, são determinados pelo mecanismo e tipo de nó. Para uma tabela desses valores de parâmetro por tipo de nó, consulte [Parâmetros específicos de node-type do Redis](#).

Tópicos

- [Gerenciamento de parâmetros](#)
- [Camadas de grupos de parâmetros de cache](#)
- [Criar um parameter group](#)
- [Listagem de grupos de parâmetros por nome](#)
- [Listagem dos valores de um grupo de parâmetros](#)
- [Modificar um parameter group](#)
- [Exclusão de um grupo de parâmetros](#)
- [Parâmetros específicos do Memcached](#)
- [Parâmetros específicos do Redis](#)

Gerenciamento de parâmetros

Os parâmetros são agrupados em parameter groups 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.
- A família do parameter groups e o cluster que você está atribuindo a ela devem ser compatíveis. Por exemplo, se seu cluster estiver executando o Redis versão 3.2.10, você só poderá usar parameter groups, padrão ou personalizados, da família Redis3.2.
- 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.
- Quando você altera parâmetros de um cluster, a alteração é aplicada no cluster imediatamente ou, com as exceções indicadas adiante, após ele ser reinicializado. 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 Redis](#).

Para obter mais informações, consulte [Reinicialização de nós](#).

Alterações de parâmetro do Redis (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster Redis (modo cluster habilitado), siga as etapas a seguir.

- activerehashing
 - bancos de dados
1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
 2. Exclua o cluster do Redis (modo cluster habilitado). Consulte [Deleting clusters](#).

3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

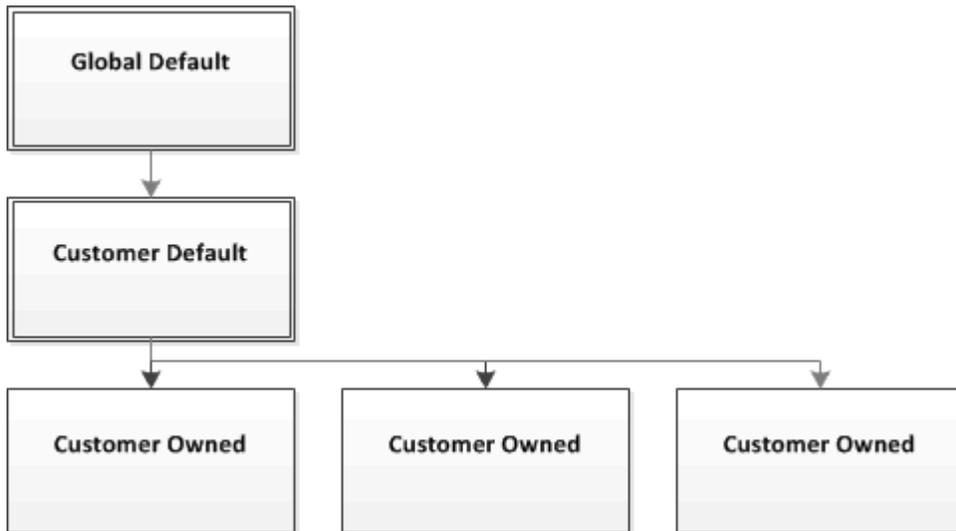
- Você pode associar grupos de parâmetros a datastores globais do Redis. Os datastores globais são uma coleção de um ou mais clusters que abrangem regiões da AWS. Nesse caso, o grupo de parâmetros é compartilhado por todos os clusters que compõem o datastore global. Quaisquer modificações no grupo de parâmetros do cluster primário são replicadas para todos os clusters restantes no datastore global. Para obter mais informações, consulte [Replicação em regiões da AWS usando o armazenamentos de dados globais](#).

Você pode verificar se um grupo de parâmetros faz parte de um datastore global procurando nestes locais:

- No console do ElastiCache, na página Parameter Groups (Grupos de parâmetros), o atributo Global yes/no (sim/não).
- A propriedade `IsGlobal` yes/no (sim/não) da operação da API [CacheParameterGroup](#)

Camadas de grupos de parâmetros de cache

O Amazon ElastiCache tem três níveis de grupos de parâmetros de cache, como mostrado a seguir.



Camadas do grupo de parâmetros do Amazon ElastiCache

Padrão global

O grupo de parâmetros raiz de nível superior para todos os clientes do Amazon ElastiCache na região.

O parameter group de cache padrão global:

- É reservado para o ElastiCache e não está disponível para o cliente.

Padrão do cliente

Uma cópia do parameter group de cache padrão global que é criado para uso do cliente.

O parameter group de cache padrão do cliente:

- É criado e de propriedade do ElastiCache.
- Está disponível ao cliente para ser usado como um parameter group de cache para quaisquer clusters executando uma versão do mecanismo com suporte por esse parameter group de cache.
- Não pode ser editado pelo cliente.

Propriedade do cliente

Uma cópia do parameter group de cache padrão do cliente. Um parameter group de cache do cliente é criado sempre que o cliente cria um parameter group de cache.

O parameter group de cache do cliente mantido:

- É criado e de propriedade do cliente.
- Pode ser atribuído a qualquer um dos clusters compatíveis com o cliente.
- Pode ser modificado pelo cliente para criar um parameter group de cache personalizado.

Nem todos os valores dos parâmetros podem ser modificados. Para obter mais informações, consulte [Parâmetros específicos do Redis](#).

Criar um parameter group

Você precisará criar um novo parameter group se houver um ou mais valores de parâmetros que você deseja alterar a partir dos valores padrão. Você pode criar um grupo de parâmetros usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Criação de um grupo de parâmetros (console)

O procedimento a seguir mostra como criar um grupo de parâmetros usando o console do ElastiCache.

Para criar um grupo de parâmetros usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Para criar um parameter group, escolha Create Parameter Group.

A tela Create Parameter Group (Criar grupo de parâmetros) será exibida.

4. Na lista Family, escolha a família de parameter groups que será o modelo para o seu parameter group.

A família de grupo de parâmetros, como redis3.2, define os parâmetros efetivos em seu grupo de parâmetros, além de seus valores iniciais. A família de parameter groups deve corresponder ao mecanismo e à versão do cluster.

5. Na caixa Name, digite um nome exclusivo para esse parameter group.

Ao criar um cluster ou modificar o parameter group de um cluster, você escolherá o parameter group pelo seu nome. Portanto, recomendamos que o nome seja informativo e de alguma forma identifique a família do parameter group.

As limitações de nomenclatura de grupo de parâmetros são as seguintes:

- Deve começar com uma letra ASCII.
- Pode conter apenas letras ASCII, dígitos e hífens.
- Deve ter entre 1 e 255 caracteres.
- Não podem conter dois hífens consecutivos.
- Não podem terminar com um hífen.

6. Na caixa Description, digite uma descrição para o parameter group.

7. Para criar o parameter group, escolha Create.

Para encerrar o processo sem criar o parameter group, escolha Cancel.

8. Quando o parameter group for criado, ele terá os valores padrão da família. Para alterar os valores padrão, você deve modificar o parameter group. Para obter mais informações, consulte [Modificar um parameter group](#).

Criação de um grupo de parâmetros (AWS CLI)

Para criar um grupo de parâmetros usando a AWS CLI, use o comando `create-cache-parameter-group` com esses parâmetros.

- `--cache-parameter-group-name`: O nome do grupo de parâmetros.

As limitações de nomenclatura de grupo de parâmetros são as seguintes:

- Deve começar com uma letra ASCII.
- Pode conter apenas letras ASCII, dígitos e hífens.
- Deve ter entre 1 e 255 caracteres.
- Não podem conter dois hífens consecutivos.
- Não podem terminar com um hífen.
- `--cache-parameter-group-family`: o mecanismo e a família de versões para o grupo de parâmetros.

- `--description`: uma descrição fornecida pelo usuário para o grupo de parâmetros.

Example

O exemplo a seguir cria um parameter group chamado `myRed28` usando a família `redis2.8` como modelo.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myRed28 \  
  --cache-parameter-group-family redis2.8 \  
  --description "My first parameter group"
```

Para Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myRed28 ^  
  --cache-parameter-group-family redis2.8 ^  
  --description "My first parameter group"
```

A saída desse comando deve ser semelhante a esta.

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myRed28",  
    "CacheParameterGroupFamily": "redis2.8",  
    "Description": "My first parameter group"  
  }  
}
```

Quando o parameter group for criado, ele terá os valores padrão da família. Para alterar os valores padrão, você deve modificar o parameter group. Para obter mais informações, consulte [Modificar um parameter group](#).

Para obter mais informações, consulte [create-cache-parameter-group](#).

Criação de um grupo de parâmetros (API do ElastiCache)

Para criar um grupo de parâmetros usando a API do ElastiCache, use a ação `CreateCacheParameterGroup` com esses parâmetros.

- `ParameterGroupName`: O nome do grupo de parâmetros.

As limitações de nomenclatura de grupo de parâmetros são as seguintes:

- Deve começar com uma letra ASCII.
 - Pode conter apenas letras ASCII, dígitos e 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, `redis2.8`.
 - `Description`: uma descrição fornecida pelo usuário para o grupo de parâmetros.

Example

O exemplo a seguir cria um parameter group chamado `myRed28` usando a família `redis2.8` como modelo.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheParameterGroup  
&CacheParameterGroupFamily=redis2.8  
&CacheParameterGroupName=myRed28  
&Description=My%20first%20parameter%20group  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

A resposta dessa ação deve ser algo semelhante ao seguinte.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <CreateCacheParameterGroupResult>  
    <CacheParameterGroup>  
      <CacheParameterGroupName>myRed28</CacheParameterGroupName>  
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>  
      <Description>My first parameter group</Description>  
    </CacheParameterGroup>  
  </CreateCacheParameterGroupResult>
```

```
<ResponseMetadata>
  <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
</ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Quando o parameter group for criado, ele terá os valores padrão da família. Para alterar os valores padrão, você deve modificar o parameter group. Para obter mais informações, consulte [Modificar um parameter group](#).

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

Listagem de grupos de parâmetros por nome

Você pode listar os grupos de parâmetros usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Listagem de grupos de parâmetros por nome (console)

O procedimento a seguir mostra como visualizar uma lista dos grupos de parâmetros usando o console do ElastiCache.

Para listar grupos de parâmetros usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.

Listagem de grupos de parâmetros por nome (AWS CLI)

Para gerar uma lista de grupos de parâmetros usando a AWS CLI, use o comando `describe-cache-parameter-groups`. Se você fornecer um nome de parameter group, somente esse parameter group será listado. Se você não fornecer o nome de um parameter group, até `--max-records` parameter groups serão listados. Em ambos os casos, o nome, a família e a descrição do parameter group estão listados.

Example

O código de exemplo a seguir lista o parameter group `myRed28`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myRed28
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myRed28
```

A saída desse comando será algo assim, listando o nome, a família e a descrição do parameter group.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed28",
      "CacheParameterGroupFamily": "redis2.8",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

O código de exemplo a seguir lista o grupo de parâmetros myRed56 para grupos de parâmetros em execução no mecanismo do Redis versão 5.0.6 e posteriores. Se o grupo de parâmetros fizer parte de um [Replicação em regiões da AWS usando o armazenamentos de dados globais](#), o valor da propriedade `IsGlobal` retornado na saída será `Yes`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name myRed56
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^
  --cache-parameter-group-name myRed56
```

A saída desse comando será algo semelhante ao seguinte: listagem de nome, família, `isGlobal` e descrição do grupo de parâmetros.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed56",
      "CacheParameterGroupFamily": "redis5.0",
      "Description": "My first parameter group",
      "IsGlobal": "yes"
    }
  ]
}
```

```
]
}
```

Example

O código de exemplo a seguir lista até 10 parameter groups.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

A saída JSON desse comando será algo semelhante ao seguinte, listagem de nome, família, descrição e, no caso do redis5.6, se o grupo de parâmetros faz parte de um datastore global (isGlobal), para cada grupo de parâmetros.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "custom-redis32",
      "CacheParameterGroupFamily": "redis3.2",
      "Description": "custom parameter group with reserved-memory > 0"
    },
    {
      "CacheParameterGroupName": "default.memcached1.4",
      "CacheParameterGroupFamily": "memcached1.4",
      "Description": "Default parameter group for memcached1.4"
    },
    {
      "CacheParameterGroupName": "default.redis2.6",
      "CacheParameterGroupFamily": "redis2.6",
      "Description": "Default parameter group for redis2.6"
    },
    {
      "CacheParameterGroupName": "default.redis2.8",
      "CacheParameterGroupFamily": "redis2.8",
      "Description": "Default parameter group for redis2.8"
    },
    {
      "CacheParameterGroupName": "default.redis3.2",
      "CacheParameterGroupFamily": "redis3.2",
      "Description": "Default parameter group for redis3.2"
    },
    {
      "CacheParameterGroupName": "default.redis3.2.cluster.on",
      "CacheParameterGroupFamily": "redis3.2",

```

```

        "Description": "Customized default parameter group for redis3.2 with
cluster mode on"
    },
    {
        "CacheParameterGroupName": "default.redis5.6.cluster.on",
        "CacheParameterGroupFamily": "redis5.0",
        "Description": "Customized default parameter group for redis5.6 with
cluster mode on",
        "isGlobal": "yes"
    },
]
}

```

Para obter mais informações, consulte [describe-cache-parameter-groups](#).

Listagem de grupos de parâmetros por nome (API do ElastiCache)

Para gerar uma lista de grupos de parâmetros usando a API do ElastiCache, use a ação `DescribeCacheParameterGroups`. Se você fornecer um nome de parameter group, somente esse parameter group será listado. Se você não fornecer o nome de um parameter group, até `MaxRecords` parameter groups serão listados. Em ambos os casos, o nome, a família e a descrição do parameter group estão listados.

Example

O código de exemplo a seguir lista até 10 parameter groups.

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&MaxRecords=10
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

A resposta dessa ação será semelhante ao seguinte, listagem do nome, família, descrição e, no caso do redis5.6, se o grupo de parâmetros pertence a um datastore (isGlobal), para cada grupo de parâmetros.

```

<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">

```

```

<DescribeCacheParameterGroupsResult>
  <CacheParameterGroups>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
      <Description>My custom Redis 2.8 parameter group</Description>
    </CacheParameterGroup>
    <CacheParameterGroup>
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
      <Description>My custom Memcached 1.4 parameter group</Description>
    </CacheParameterGroup>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
      <Description>My custom redis 5.6 parameter group</Description>
      <isGlobal>yes</isGlobal>
    </CacheParameterGroup>
  </CacheParameterGroups>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
  <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>

```

Example

O código de exemplo a seguir lista o parameter group `myRed28`.

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

A resposta dessa ação será semelhante ao seguinte: listagem do nome, família e descrição.

```

<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>

```

```

<CacheParameterGroups>
  <CacheParameterGroup>
    <CacheParameterGroupName>myRed28</CacheParameterGroupName>
    <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
    <Description>My custom Redis 2.8 parameter group</Description>
  </CacheParameterGroup>
</CacheParameterGroups>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
  <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>

```

Example

O código de exemplo a seguir lista o grupo de parâmetros myRed56.

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myRed56
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>

```

A resposta desta ação será semelhante ao seguinte, listagem do nome, família, descrição e se o grupo de parâmetros faz parte de um datastore global (isGlobal).

```

<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom Redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>

```

```
</ResponseMetadata>  
</DescribeCacheParameterGroupsResponse>
```

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

Listagem dos valores de um grupo de parâmetros

Você pode listar os parâmetros e seus valores para um grupo de parâmetros usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Listagem dos valores de um grupo de parâmetros (console)

O procedimento a seguir mostra como listar os parâmetros e seus valores para um grupo de parâmetros usando o console do ElastiCache.

Para listar os parâmetros de um grupo e seus valores usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Escolha o parameter group para o qual você deseja listar os parâmetros e os valores, escolhendo a caixa à esquerda do nome do parameter group.

Os parâmetros e seus valores serão listados na parte inferior da tela. Devido ao número de parâmetros, talvez seja necessário rolar para cima e para baixo para encontrar o parâmetro de interesse.

Listagem dos valores de um grupo de parâmetros (AWS CLI)

Para listar os parâmetros de um grupo de parâmetros e seus valores usando a AWS CLI, use o comando `describe-cache-parameters`.

Example

O código de exemplo a seguir lista todos os parâmetros e seus valores para o parameter group `myRedis28`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myRedis28
```

Para Windows:

```
aws elasticache describe-cache-parameters ^
```

```
--cache-parameter-group-name myRed28
```

Para obter mais informações, consulte [describe-cache-parameters](#).

Listagem dos valores de um grupo de parâmetros (API do ElastiCache)

Para listar os parâmetros de um grupo de parâmetros e seus valores usando a API do ElastiCache, use a ação DescribeCacheParameters.

Example

O código de exemplo a seguir lista todos os parâmetros para o parameter group myRed28.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameters  
&CacheParameterGroupName=myRed28  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

A resposta dessa ação será semelhante a esta. Essa resposta foi truncada.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParametersResult>  
    <CacheClusterClassSpecificParameters>  
      <CacheNodeTypeSpecificParameter>  
        <DataType>integer</DataType>  
        <Source>system</Source>  
        <IsModifiable>>false</IsModifiable>  
        <Description>The maximum configurable amount of memory to use to store items,  
in megabytes.</Description>  
        <CacheNodeTypeSpecificValues>  
          <CacheNodeTypeSpecificValue>  
            <Value>1000</Value>  
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>  
          <CacheNodeTypeSpecificValue>  
            <Value>6000</Value>  
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>
```

```
<CacheNodeTypeSpecificValue>
  <Value>7100</Value>
  <CacheClusterClass>cache.m1.large</CacheClusterClass>
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>1300</Value>
  <CacheClusterClass>cache.m1.small</CacheClusterClass>
</CacheNodeTypeSpecificValue>

...output omitted...

</CacheClusterClassSpecificParameters>
</DescribeCacheParametersResult>
<ResponseMetadata>
  <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParametersResponse>
```

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

Modificar um parameter group

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 alteração no valor de um parâmetro é aplicada a um parameter group, consulte [Parâmetros específicos do Redis](#).

Modificação de um grupo de parâmetros (console)

O procedimento a seguir mostra como alterar o valor do parâmetro `cluster-enabled` usando o console do ElastiCache. Você usaria o mesmo procedimento para alterar o valor de qualquer parâmetro.

Para alterar o valor de um parâmetro usando o console do ElastiCache

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

2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Escolha o parameter group que você deseja modificar, marcando a caixa à esquerda do nome do parameter group.

Os parâmetros do parameter group serão listados na parte inferior da tela. Talvez você precise pular pela lista para ver todos os parâmetros.

4. Para modificar um ou mais parâmetros, escolha Edit Parameters.
5. Escolha Save Changes (Salvar alterações).
6. Para localizar o parâmetro que você alterou, consulte [Parâmetros específicos do Redis](#). Se você tiver um cluster do Redis (modo cluster desabilitado) e fizer alterações nos parâmetros a seguir, reinicialize os nós no cluster:

- activerehashing
- bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

Alterações de parâmetro do Redis (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster Redis (modo cluster habilitado), siga as etapas a seguir.

- activerehashing
 - bancos de dados
1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
 2. Exclua o cluster do Redis (modo cluster habilitado). Consulte [Deleting clusters](#).
 3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Modificação de um grupo de parâmetros (AWS CLI)

Para alterar o valor de um parâmetro usando a AWS CLI, use o comando `modify-cache-parameter-group`.

Example

Para encontrar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros específicos do Redis](#)

O código de exemplo a seguir define o valor de dois parâmetros, `reserved-memory-percent` e `cluster-enabled` no parameter group `myredis32-on-30`. Definimos `reserved-memory-percent` como `30` (30%) e `cluster-enabled` como `yes` para que o grupo de parâmetros possa ser usado com os clusters do Redis (modo cluster habilitado) (grupos de replicação).

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myredis32-on-30 \  
  --parameter-name-values \  
    ParameterName=reserved-memory-percent,ParameterValue=30 \  
    ParameterName=cluster-enabled,ParameterValue=yes
```

Para Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myredis32-on-30 ^  
  --parameter-name-values ^  
    ParameterName=reserved-memory-percent,ParameterValue=30 ^  
    ParameterName=cluster-enabled,ParameterValue=yes
```

A saída desse comando será semelhante a esta.

```
{  
  "CacheParameterGroupName": "my-redis32-on-30"  
}
```

Para obter mais informações, consulte [modify-cache-parameter-group](#).

Para localizar o parâmetro que você alterou, consulte [Parâmetros específicos do Redis](#).

Se você tiver um cluster do Redis (modo cluster desabilitado) e fizer alterações nos parâmetros a seguir, reinicialize os nós no cluster:

- `activeresharding`
- bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

Alterações de parâmetro do Redis (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster Redis (modo cluster habilitado), siga as etapas a seguir.

- `activeresharding`
 - bancos de dados
1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
 2. Exclua o cluster do Redis (modo cluster habilitado). Consulte [Deleting clusters](#).
 3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Modificação de um grupo de parâmetros (API do ElastiCache)

Para alterar os valores dos parâmetros de um grupo usando a API do ElastiCache, use a ação `ModifyCacheParameterGroup`.

Example

Para encontrar o nome e os valores permitidos do parâmetro que você deseja alterar, consulte [Parâmetros específicos do Redis](#)

O código de exemplo a seguir define o valor de dois parâmetros, `reserved-memory-percent` e `cluster-enabled` no parameter group `myredis32-on-30`. Definimos `reserved-memory-percent` como `30` (30%) e `cluster-enabled` como `yes` para que o grupo de parâmetros possa ser usado com os clusters do Redis (modo cluster habilitado) (grupos de replicação).

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheParameterGroup  
&CacheParameterGroupName=myredis32-on-30  
&ParameterNameValues.member.1.ParameterName=reserved-memory-percent  
&ParameterNameValues.member.1.ParameterValue=30  
&ParameterNameValues.member.2.ParameterName=cluster-enabled  
&ParameterNameValues.member.2.ParameterValue=yes  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

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

Se você tiver um cluster do Redis (modo cluster desabilitado) e fizer alterações nos parâmetros a seguir, reinicialize os nós no cluster:

- activerehashing
- bancos de dados

Para obter mais informações, consulte [Reinicialização de nós](#).

Alterações de parâmetro do Redis (modo cluster habilitado)

Se você fizer alterações nos parâmetros a seguir em um cluster Redis (modo cluster habilitado), siga as etapas a seguir.

- activerehashing
 - bancos de dados
1. Crie um backup manual do seu cluster. Consulte [Realização de backups manuais](#).
 2. Exclua o cluster do Redis (modo cluster habilitado). Consulte [Excluir um cluster](#).
 3. Restaure o cluster usando o parameter group alterado e o backup para propagar o novo cluster. Consulte [Restauração de um backup para um novo cache](#).

Alterações em outros parâmetros não exigem isso.

Exclusão de um grupo de parâmetros

Você pode excluir um grupo de parâmetros personalizado usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Não será possível excluir um parameter group se ele estiver associado a qualquer cluster. Você também não pode excluir nenhum dos parameter groups padrão.

Exclusão de um grupo de parâmetros (console)

O procedimento a seguir mostra como excluir um grupo de parâmetros usando o console do ElastiCache.

Para excluir um grupo de parâmetros usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os parameter groups disponíveis, no painel de navegação à esquerda, escolha Parameter Groups.
3. Escolha os parameter groups que você deseja excluir marcando a caixa à esquerda do nome do parameter group.

O botão Delete ficará ativo.

4. Escolha Delete (Excluir).

A tela de confirmação Delete Parameter Groups será exibida.

5. Para excluir os parameter groups, na tela de confirmação Delete Parameter Groups, escolha Delete.

Para manter os parameter groups, escolha Cancel.

Exclusão de um grupo de parâmetros (AWS CLI)

Para excluir um grupo de parâmetros usando a AWS CLI, use o comando `delete-cache-parameter-group`. Para o parameter group a ser excluído, o parameter group especificado por `--cache-parameter-group-name` não pode ter nenhum cluster associado a ele, nem pode ser um parameter group padrão.

O código de exemplo a seguir exclui o parameter group `myMem14`.

Example

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myRed28
```

Para Windows:

```
aws elasticache delete-cache-parameter-group ^  
  --cache-parameter-group-name myRed28
```

Para obter mais informações, consulte [delete-cache-parameter-group](#).

Exclusão de um grupo de parâmetros (API do ElastiCache)

Para excluir um grupo de parâmetros usando a API do ElastiCache, use a ação `DeleteCacheParameterGroup`. Para o parameter group a ser excluído, o parameter group especificado por `CacheParameterGroupName` não pode ter nenhum cluster associado a ele, nem pode ser um parameter group padrão.

Example

O código de exemplo a seguir exclui o parameter group `myRed28`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myRed28  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

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

Parâmetros específicos do 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 [Criar um `parameter group`](#).

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
no_modern	<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 os comandos <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> . O <code>No modern</code> também define o <code>hash_algorithm</code> para o <code>jenkins</code> e permite a incorporação do VALOR ASCII. Aplicável para versões 1.5 e superiores do memcached. Para reverter para o moderno, é necessário desabilitar esse parâmetro e reiniciar, o que habilitará automaticamente <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> e <code>maxconns_fast</code> .</p> <div data-bbox="1003 1220 1510 1885"><p> Note</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 obtido automaticamente por novos usuários do elasticache para cada região após 20 de agosto de 2021. Os usuários do ElastiCache existentes nas</p></div>

Nome	Detalhes	Descrição
		regiões antes de 20 de agosto de 2021 precisam modificar manualmente seus grupos de parâmetros personalizados para obter essa nova alteração.
<code>inline_ascii_resp</code>	<p>Padrão: 0</p> <p>Tipo: booleano</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 os conjuntos do ASCII get, faster.

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: booleano</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>	Padrão: 1	

Nome	Detalhes	Descrição
	<p>Tipo: booleano</p> <p>Modificável: sim (requer reinicialização se definido para <code>no_modern</code>)</p> <p>Valores permitidos: 0,1</p> <p>As alterações terão efeito: na execução</p>	<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.</p>

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
<code>modern</code>	<p>Padrão: habilitado</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: na execução</p>	<p>Um alias para vários recursos. Habilitar <code>modern</code> 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> .</p>

Nome	Detalhes	Descrição
<code>watch</code>	<p>Padrão: habilitado</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <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>	<p>Buscas, remoções ou mutações de logs. Quando, por exemplo, o usuário ativa <code>watch</code>, ele pode ver logs quando ocorre <code>get</code>, <code>set</code>, <code>delete</code> ou <code>update</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: booleano</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;"><p> Note</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 1797" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</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 Modificar um parameter group.</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 thread em segundo plano que embaralha itens entre as LRUs à 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> Warning</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 configuração do ElastiCache.
<code>config_size_max</code>	O tamanho máximo das entradas de configuração, em bytes.
<code>hashpower_init</code>	O tamanho inicial da tabela de hash do ElastiCache, 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 estiver definido como 1, o ElastiCache enviará um erro ao cliente e fechará 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, o ElastiCache usará uma abordagem lenta e conservadora para mover slabs automaticamente. Se estiver definido como 2, o ElastiCache moverá os slabs agressivamente sempre que houver uma remoção. (Esse modo não é recomendado, exceto para fins de teste.)</p>

Nome	Descrição
slab_reassign	Habilite ou desabilite a reatribuição de slabs. Se esse parâmetro estiver definido como 1, você poderá usar o comando "slabs reassign" para reatribuir a memória manualmente.

Parâmetros com suporte do Memcached 1.4.5

Família de parameter groups: memcached1.4

Para o Memcached 1.4.5, há suporte para os seguintes parâmetros.

Parâmetros adicionados no Memcached 1.4.5

Nome	Detalhes	Descrição
backlog_queue_limit	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 Modificar um parameter group.</p>
<code>cas_disabled</code>	<p>Padrão: 0 (false)</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	<p>Se 1 (true), as operações Verificar e Definir (CAS) serão desabilitadas, e os itens armazenados consumirão 8 bytes a menos do que quando essas operações estão habilitadas.</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>	<p>Padrão: 0 (false)</p> <p>Tipo: booliano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	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>	<p>Padrão: 0 (false)</p> <p>Tipo: booliano</p> <p>Modificável: não</p>	Se 1 (true), o ElastiCache tentará usar páginas de memória grandes.
<code>lock_down_paged_memory</code>	<p>Padrão: 0 (false)</p> <p>Tipo: booliano</p> <p>Modificável: não</p>	Se 1 (true), o ElastiCache bloqueará toda a memória paginada.
<code>max_item_size</code>	<p>Padrão: 1048576</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p>	O tamanho, em bytes, do maior item que pode ser armazenado no cluster.
<code>max_simultaneous_connections</code>	<p>Padrão: 65000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	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 (true), o ElastiCache maximizará o limite do arquivo core.
<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 Sobrecarga de conexões do Memcached .
<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 seu cluster estiver usando o grupo de parâmetros padrão, o ElastiCache definirá o valor para `memcached_connections_overhead` como 13 MB.
- Se o seu cluster estiver usando um `parameter group` que você criou, será possível definir o valor de `memcached_connections_overhead` como um valor de sua escolha.

Parâmetros específicos do tipo de nó do Memcached

Embora a maioria dos parâmetros tenha um valor único, alguns parâmetros têm valores diferentes dependendo do tipo de nó usado. A tabela a seguir mostra os valores padrão para os parâmetros `max_cache_memory` e `num_threads` para cada tipo de nó. Os valores nesses parâmetros não podem ser modificados.

Tipo de nó	<code>max_cache_memory</code> (em megabytes)	<code>num_threads</code>
<code>cache.t1.micro</code>	213	1
<code>cache.t2.micro</code>	555	1
<code>cache.t2.small</code>	1588	1
<code>cache.t2.medium</code>	3301	2
<code>cache.t3.micro</code>	512	2

Tipo de nó	max_cache_memory (em megabytes)	num_threads
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2
cache.m2.2xlarge	30412	4
cache.m2.4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4
cache.m3.2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4
cache.m4.2xlarge	30412	8
cache.m4.4xlarge	62234	16

Tipo de nó	max_cache_memory (em megabytes)	num_threads
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8
cache.r3.4xlarge	120600	16
cache.r3.8xlarge	120600	32

Tipo de nó	max_cache_memory (em megabytes)	num_threads
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
cache.r4.4xlarge	103815	16
cache.r4.8xlarge	208144	32
cache.r4.16xlarge	416776	64
cache.r5.large	13387	2
cache.r5.xlarge	26953	4
cache.r5.2xlarge	54084	8
cache.r5.4xlarge	108347	16
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48
cache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2

Tipo de nó	max_cache_memory (em megabytes)	num_threads
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

Note

Todas as instâncias T2 são criadas em uma Amazon Virtual Private Cloud (Amazon VPC).

Parâmetros específicos do Redis

Se você não especificar um grupo de parâmetros para seu cluster do Redis, será usado um grupo de parâmetros padrão apropriado à versão de seu mecanismo. Não é possível alterar os valores de nenhum parâmetro em um grupo de parâmetros padrão. No entanto, é possível criar um grupo de parâmetros personalizado e atribuí-lo ao seu cluster a qualquer momento, desde que os valores de parâmetros condicionalmente modificáveis sejam os mesmos nos dois grupos de parâmetros. Para obter mais informações, consulte [Criar um parameter group](#).

Tópicos

- [Alterações de parâmetros do Redis 7](#)
- [Alterações de parâmetros do Redis 6.x](#)
- [Alterações de parâmetros do Redis 5.0.3](#)
- [Alterações de parâmetros do Redis 5.0.0](#)
- [Alterações de parâmetros do Redis 4.0.10](#)
- [Alterações de parâmetros do Redis 3.2.10](#)
- [Alterações de parâmetros do Redis 3.2.6](#)
- [Alterações de parâmetros do Redis 3.2.4](#)
- [Parâmetros adicionados do Redis 2.8.24 \(aprimorado\)](#)
- [Parâmetros adicionados do Redis 2.8.23 \(aprimorado\)](#)
- [Parâmetros adicionados do Redis 2.8.22 \(aprimorado\)](#)
- [Parâmetros adicionados do Redis 2.8.21](#)
- [Parâmetros adicionados do Redis 2.8.19](#)
- [Parâmetros adicionados do Redis 2.8.6](#)
- [Parâmetros do Redis 2.6.13](#)
- [Parâmetros específicos de node-type do Redis](#)

Alterações de parâmetros do Redis 7

Família de grupo de parâmetros: redis7

Os grupos de parâmetros padrão do Redis 7 são os seguintes:

- `default.redis7`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster desabilitado) e grupos de replicação.

- `default.redis7.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster habilitado) e grupos de replicação.

Os parâmetros adicionados no Redis 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 solicitações <code>MOVED/ASKING</code>, bem como o campo do endpoint para <code>CLUSTER SLOTS</code> e <code>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 a criptografia em trânsito estiver habilitada e um endereço IP, caso contrário.</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>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
	As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	
hash-max-listpack-entries	Valores permitidos: 0+ Padrão: 512 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O número máximo de entradas de hash para que o conjunto de dados seja compactado.
hash-max-listpack-value	Valores permitidos: 0+ Padrão: 64 Tipo: inteiro Modificável: sim As alterações entrarão em vigor: imediatamente em todos os nós no cluster.	O limite das maiores entradas de hash para que o conjunto de dados seja compactado.

Nome	Detalhes	Descrição
<code>zset-max-listpack-entries</code>	<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>	O número máximo de entradas do conjunto classificado para que o conjunto de dados seja compactado.
<code>zset-max-listpack-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>	O limite das maiores entradas do conjunto classificado para que o conjunto de dados seja compactado.

Os parâmetros alterados no Redis 7 são os seguintes.

Nome	Detalhes	Descrição
<code>activeresharding</code>	<p>Permite modificação: no. No Redis 7, esse parâmetro está oculto e ativado por padrão. Para desativá-lo, você precisa criar um caso de suporte.</p>	Permite modificação era Sim.

Os parâmetros removidos no Redis 7 são os seguintes.

Nome	Detalhes	Descrição
<code>hash-max-ziplist-entries</code>	<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>	Use <code>listpack</code> em vez de <code>ziplist</code> para representar uma pequena codificação de hash
<code>hash-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>zset-max-ziplist-entries</code>	<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>	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 de parâmetros do Redis 6.x

Família de grupo de parâmetros: `redis6.x`

Os grupos de parâmetros padrão do Redis 6.x são os seguintes:

- `default.redis6.x`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster desabilitado) e grupos de replicação.
- `default.redis6.x.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster habilitado) e grupos de replicação.

Note

No mecanismo Redis versão 6.2, quando a família de nós r6gd foi introduzida para uso com [Classificação de dados em níveis](#), somente as políticas max-memory noeviction, volatile-lru and allkeys-lru são compatíveis com os tipos de nó r6gd.

Para obter mais informações, consulte [ElastiCache para Redis versão 6.2 \(aprimorado\)](#) e [ElastiCache para Redis versão 6.0 \(aprimorado\)](#).

Os parâmetros adicionados no Redis 6.x são os seguintes.

Nome	Detalhes	Descrição
acl-pubsub-default (added in 6.2)	<p>Valores permitidos: resetchannels , allchannels</p> <p>Padrão: allchannels</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>Entrada das alterações em vigor: os usuários existentes do Redis associados ao cluster continuarão a ter as permissões existentes. Atualize os usuários ou reinicialize o cluster para atualizar os usuários existentes do Redis.</p>	Permissões de canal pubsub padrão para usuários de ACL implantados neste cluster.
cluster-allow-reads-when-do	<p>Padrão: não</p> <p>Tipo: string</p>	Quando definido como yes, um grupo de replicação do Redis (modo cluster habilitado) continua a processar comandos de leitura

Nome	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>mesmo quando um nó não é capaz de alcançar um 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 ajudar o cache do lado do cliente, o Redis oferece suporte ao monitoramento 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>

Nome	Detalhes	Descrição
<code>acllog-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 log da ACL.</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 exclui chaves que excederam seu tempo de vida por dois mecanismos. Em um, uma chave é acessada e se descobre que ela está expirada. No outro, um trabalho periódico amostra as chaves e faz com que aquelas que excederam seu tempo de vida expirem. Esse parâmetro define a quantidade de esforço que o Redis 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 compensação é mais CPU e latência potencialmente maior. Recomendamos um valor de 1, a menos que você esteja vendo alto uso de memória e possa tolerar um aumento na utilização da CPU.</p>

Nome	Detalhes	Descrição
lazyfree-lazy-user-del (added in 6.0)	<p>Padrão: não</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações entrarão em vigor: imediatamente em todos os nós no cluster.</p>	Quando o valor é definido como yes (sim), o comando DEL funciona da mesma forma que UNLINK.

Os parâmetros removidos no Redis 6.x são os seguintes.

Nome	Detalhes	Descrição
lua-replique-commands	<p>Valores permitidos: sim/não</p> <p>Padrão: sim</p> <p>Tipo: booliano</p> <p>Modificável: sim</p> <p>As alterações entram em vigor: imediatamente</p>	Sempre habilite ou não a replicação de efeito Lua em scripts Lua

Alterações de parâmetros do Redis 5.0.3

Família de grupo de parâmetros: redis5.0

Grupos de parâmetros padrão do Redis 5.0

- `default.redis5.0`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster desabilitado) e grupos de replicação.
- `default.redis5.0.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster habilitado) e grupos de replicação.

Parâmetros adicionados no Redis 5.0.3

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 Redis renomeados. 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 para Redis versão 5.0.6 \(aprimorado\)](#).

Alterações de parâmetros do Redis 5.0.0

Família de grupo de parâmetros: redis5.0

Grupos de parâmetros padrão do Redis 5.0

- `default.redis5.0`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster desabilitado) e grupos de replicação.
- `default.redis5.0.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster habilitado) e grupos de replicação.

Parâmetros adicionados no Redis 5.0

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
<code>lua-replicate-commands</code>	Valores permitidos: sim/não Padrão: sim Tipo: booliano Modificável: sim As alterações entram em vigor: imediatamente	Sempre habilite ou não a replicação de efeito Lua em scripts Lua
<code>replica-ignore-maxmemory</code>	Padrão: sim Tipo: booliano Modificável: não	Determina se a réplica ignora a configuração <code>maxmemory</code> ao não remover itens independentes do primário

O Redis renomeou vários parâmetros na versão 5.0 do mecanismo em resposta ao feedback da comunidade. Para obter mais informações, consulte [O que há de novo no Redis 5?](#) A tabela a seguir lista os novos nomes e como eles mapeiam para versões anteriores.

Parâmetros renomeados no Redis 5.0

Nome	Detalhes	Descrição
<code>replica-lazy-flush</code>	Padrão: sim Tipo: booliano Modificável: não Nome anterior: <code>slave-lazy-flush</code>	Realiza um <code>flushDB</code> assíncrono durante a sincronização da réplica.
<code>client-output-buffer</code>	Padrão: para valores, consulte Parâmetros específicos de node-type do Redis	Para réplicas de leitura do Redis: se o buffer de saída de um cliente atingir o número especificado de bytes, o cliente será desconectado.

Nome	Detalhes	Descrição
<code>er-limit-replica-hard-limit</code>	<p>Tipo: inteiro</p> <p>Modificável: não</p> <p>Nome anterior: <code>client-output-buffer-limit-slave-hard-limit</code></p>	
<code>client-output-buffer-limit-replica-soft-limit</code>	<p>Padrão: para valores, consulte Parâmetros específicos de node-type do Redis</p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>Nome anterior: <code>client-output-buffer-limit-slave-soft-limit</code></p>	Para réplicas de 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 por <code>client-output-buffer-limit-replica-soft-seconds</code> .
<code>client-output-buffer-limit-replica-soft-seconds</code>	<p>Padrão: 60</p> <p>Tipo: inteiro</p> <p>Modificável: não</p> <p>Nome anterior: <code>client-output-buffer-limit-slave-soft-seconds</code></p>	Para réplicas de leitura do Redis: se o buffer de saída de um cliente permanecer em <code>client-output-buffer-limit-replica-soft-limit</code> bytes por mais tempo que esse número de segundos, o cliente será desconectado.
<code>replica-allow-chaining</code>	<p>Padrão: não</p> <p>Tipo: string</p> <p>Modificável: não</p> <p>Nome anterior: <code>slave-allow-chaining</code></p>	Determina se uma réplica de leitura no Redis pode ter réplicas de leitura próprias.

Nome	Detalhes	Descrição
<code>min-replicas-to-write</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>Nome anterior: <code>min-slaves-to-write</code></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, então o nó primário sempre aceitará solicitações de gravação, mesmo que não haja réplicas disponíveis.</p>
<code>min-replicas-max-lag</code>	<p>Padrão: 10</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>Nome anterior: <code>min-slaves-max-lag</code></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 de <code>min-replicas-to-write</code>, então o primário deixará de aceitar gravações nesse ponto.</p> <p>Se esse parâmetro ou <code>min-replicas-to-write</code> for 0, então o nó primário sempre aceitará solicitações de gravação, mesmo que não haja réplicas disponíveis.</p>

Nome	Detalhes	Descrição
<code>close-on-replica-write</code>	Padrão: sim Tipo: booleano Modificável: sim Nome anterior: <code>close-on-slave-write</code> As alterações terão efeito: imediatamente	Se ativado, os clientes que tentarem gravar em uma réplica somente leitura serão desconectados.

Parâmetros removidos no Redis 5.0

Nome	Detalhes	Descrição
<code>repl-timeout</code>	Padrão: 60 Modificável: não	O parâmetro não está disponível nesta versão.

Alterações de parâmetros do Redis 4.0.10

Família de grupo de parâmetros: `redis4.0`

Grupos de parâmetros padrão do Redis 4.0.x

- `default.redis4.0`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster desabilitado) e grupos de replicação.
- `default.redis4.0.cluster.on`: use este grupo de parâmetros, ou um grupo derivado dele, para clusters do Redis (modo cluster habilitado) e grupos de replicação.

Parâmetros alterados no Redis 4.0.10

Nome	Detalhes	Descrição
maxmemory-policy	<p>Valores permitidos: allkeys-lru , volatile-lru , allkeys-lfu , volatile-lfu , allkeys-random , volatile-random , volatile-ttl , noeviction</p> <p>Padrão: volatile-lru</p> <p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	<p>maxmemory-policy foi adicionado na versão 2.6.13. Na versão 4.0.10 foram adicionados dois novos valores permitido s: allkeys-lfu , que excluirá qualquer chave que esteja usando LFU aproximado, e volatile-lfu , que excluirá o uso de LFU aproximado entre as chaves com um conjunto de expiração. Na versão 6.2, quando a família de nós r6gd foi introduzida para uso com classificação de dados em níveis, somente as políticas noeviction , volatile-lru e allkeys-lru são compatíveis com os tipos de nó r6gd.</p>

Parâmetros adicionados no Redis 4.0.10

Nome	Detalhes	Descrição
Parâmetros de exclusão assíncrona		
lazyfree-lazy- eviction	<p>Valores permitidos: sim/não</p> <p>Padrão: não</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	<p>Performs an asynchronous delete on evictions.</p>

Nome	Detalhes	Descrição
<code>lazyfree-lazy-expire</code>	Valores permitidos: sim/não Padrão: não Tipo: booleano Modificável: sim As alterações ocorrem: imediatamente	Performs an asynchronous delete on expired keys.
<code>lazyfree-lazy-server-del</code>	Valores permitidos: sim/não Padrão: não Tipo: booleano Modificável: sim As alterações ocorrem: imediatamente	Performs an asynchronous delete for commands which update values.
<code>slave-lazy-flush</code>	Valores permitidos: N/D Padrão: não Tipo: booleano Modificável: não Changes take place: N/A	Performs an asynchronous flushDB during slave sync.

Parâmetros de LFU

Nome	Detalhes	Descrição
<code>lfu-log-factor</code>	<p>Valores permitidos: qualquer inteiro > 0</p> <p>Padrão: 10</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	<p>Set the log factor, which determines the number of key hits to saturate the key counter.</p>
<code>lfu-decay-time</code>	<p>Valores permitidos: qualquer inteiro</p> <p>Padrão: 1</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	<p>The amount of time in minutes to decrement the key counter.</p>
Parâmetros de desfragmentação ativa		
<code>activedefrag</code>	<p>Valores permitidos: sim/não</p> <p>Padrão: não</p> <p>Tipo: booleano</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	<p>Enabled active defragmentation.</p>

Nome	Detalhes	Descrição
active-defrag-ignore-bytes	Valores permitidos: 10485760-104857600 Padrão: 104857600 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Minimum amount of fragmentation waste to start active defrag.
active-defrag-threshold-lower	Valores permitidos: 1-100 Padrão: 10 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Minimum percentage of fragmentation to start active defrag.
active-defrag-threshold-upper	Valores permitidos: 1-100 Padrão: 100 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Maximum percentage of fragmentation at which we use maximum effort.

Nome	Detalhes	Descrição
<code>active-defrag-cycle-min</code>	<p>Valores permitidos: 1-75</p> <p>Padrão: 25</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	Minimal effort for defrag in CPU percentage.
<code>active-defrag-cycle-max</code>	<p>Valores permitidos: 1-75</p> <p>Padrão: 75</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	Maximal effort for defrag in CPU percentage.
Parâmetros de buffer de saída do cliente		
<code>client-query-buffer-limit</code>	<p>Valores permitidos: 1048576-1073741824</p> <p>Padrão: 1073741824</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações ocorrem: imediatamente</p>	Max size of a single client query buffer.

Nome	Detalhes	Descrição
<code>proto-max-bulk-len</code>	Valores permitidos: 1048576-536870912 Padrão: 536870912 Tipo: inteiro Modificável: sim As alterações ocorrem: imediatamente	Max size of a single element request.

Alterações de parâmetros do Redis 3.2.10

Família de grupo de parâmetros: `redis3.2`

Para o ElastiCache para Redis 3.2.10 não há parâmetros adicionais com suporte.

Alterações de parâmetros do Redis 3.2.6

Família de grupo de parâmetros: `redis3.2`

Para o Redis 3.2.6 não há parâmetros adicionais com suporte.

Alterações de parâmetros do Redis 3.2.4

Família de grupo de parâmetros: `redis3.2`

Começando com o Redis 3.2.4 existem dois grupo de parâmetros padrão.

- `default.redis3.2`: ao executar o Redis 3.2.4, especifique este grupo de parâmetros ou um derivado dele se quiser criar um grupo de replicação do Redis (modo cluster desabilitado) e ainda usar os recursos adicionais do Redis 3.2.4.
- `default.redis3.2.cluster.on`: especifique este grupo de parâmetros ou um grupo de parâmetros derivado dele, quando quiser criar um grupo de replicação do Redis (modo cluster habilitado).

Tópicos

- [Novos parâmetros para o Redis 3.2.4](#)
- [Parâmetros alterados no Redis 3.2.4 \(aprimorado\)](#)

Novos parâmetros para o Redis 3.2.4

Família de grupo de parâmetros: redis3.2

Para o Redis 3.2.4 há suporte para os seguintes parâmetros adicionais.

Nome	Detalhes	Descrição
<code>list-max-ziplist-size</code>	<p>Padrão: -2</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	<p>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 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"> • -5: tamanho máximo: 64 Kb - não recomendado para workloads normais • -4: tamanho máximo: 32 Kb - não recomendado • -3: tamanho máximo: 16 Kb - não recomendado • -2: tamanho máximo: 8 Kb - recomendado • -1: tamanho máximo: 4 Kb - recomendado • Números positivos significam armazenar exatamente o número de elementos por nó de lista.

Nome	Detalhes	Descrição
list-compress-depth	<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"> • 0: desabilitar toda a compactação. • 1: começar a compactar com o 1º nó de início e final. [início] -> nó-> nó -> ...-> nó -> [final] Todos os nós, exceto [início] e [final] são compactados. • 2: começar a compactar com o 2º nó de início e final. [início] -> [próximo] -> nó-> nó -> ...-> nó -> [anterior] -> [final] [início], [próximo], [anterior], [final] não são compactados. Todos os outros nós são compactados. • Etc.

Nome	Detalhes	Descrição
<code>cluster-enabled</code>	<p>Padrão: não/sim*</p> <p>Tipo: string</p> <p>Modificável: não</p>	<p>Indica se este é um grupo de replicação do Redis (modo cluster habilitado) no modo de cluster (sim) ou um grupo de replicação do Redis (modo cluster habilitado) no modo sem cluster (não). Grupos de replicação do Redis (modo cluster habilitado) no modo de cluster podem particionar seus dados em até 500 grupos de nós.</p> <p>*O Redis 3.2.x tem dois grupo de parâmetros padrão.</p> <ul style="list-style-type: none">• <code>default.redis3.2</code> – valor padrão no.• <code>default.redis3.2.cluster.on</code> – valor padrão yes. <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 estão definidos como <code>yes</code>, os nós do Redis (modo cluster habilitado) no modo de cluster param de aceitar consultas quando detectam pelo menos um slot de hash descoberto (nenhum nó disponível que o sirva). Dessa forma, se o cluster estiver parcialmente inativo, ele ficará indisponível. Ele voltará a ficar disponível automaticamente assim que todos os slots forem novamente cobertos.</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>Limite de bytes de representação esparsa de HyperLogLog. O limite inclui o cabeçalho de 16 bytes. Quando um HyperLogLog usando a representação esparsa ultrapassar esse limite, ele será 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 3000 para ter os benefícios da codificação eficiente do espaço sem diminuir demais o PFADD, que é $O(N)$ com a codificação esparsa. O valor pode ser aumentado para ~10000 quando a CPU não é uma preocupação, mas o espaço sim, e o conjunto de dados for composto por muitos HyperLogLogs com cardinalidade no intervalo 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 dados do Redis cresce até consumir toda a memória do nó. Se isso ocorrer, o desempenho do nó provavelmente sofrerá devido a uma paginação de memória excessiva. Ao reservar memória, você pode separar uma parte da memória disponível para fins não Redis, a fim de ajudar a reduzir a quantidade de paginação.</p> <p>Este parâmetro é específico do ElastiCache e não faz parte da distribuição padrão do Redis.</p> <p>Para obter mais informações, consulte <code>reserved-memory</code> e Gerenciamento da memória reservada.</p>

Parâmetros alterados no Redis 3.2.4 (aprimorado)

Família de grupo de parâmetros: redis3.2

Para o Redis 3.2.4 os parâmetros a seguir 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 associado a nenhum cluster de cache. Do contrário, não.	Modificável era Não.

Nome	Detalhes	Alteração
appendonly	Padrão: desativado Modificável: não	Se quiser atualizar de uma versão Redis anterior, primeiro você deve desativar <code>appendonly</code> .
appendfsync	Padrão: desativado Modificável: não	Se quiser atualizar de uma versão Redis anterior, primeiro você deve desativar <code>appendfsync</code> .
repl-timeout	Padrão: 60 Modificável: não	Agora, ele não é modificável com um padrão de 60.
tcp-keepalive	Padrão: 300	O padrão era 0.
list-max-ziplist-entries		O parâmetro não está mais disponível.
list-max-ziplist-value		O parâmetro não está mais disponível.

Parâmetros adicionados do Redis 2.8.24 (aprimorado)

Família de grupo de parâmetros: `redis2.8`

Para o Redis 2.8.24 não há parâmetros adicionais com suporte.

Parâmetros adicionados do Redis 2.8.23 (aprimorado)

Família de grupo de parâmetros: `redis2.8`

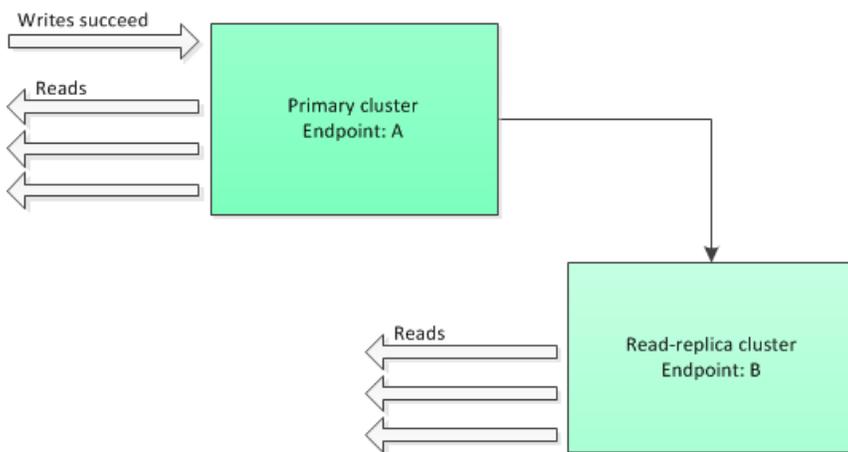
Para o Redis 2.8.23 o parâmetro adicional a seguir tem suporte.

Nome	Detalhes	Descrição
<code>close-on-slave-write</code>	<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.

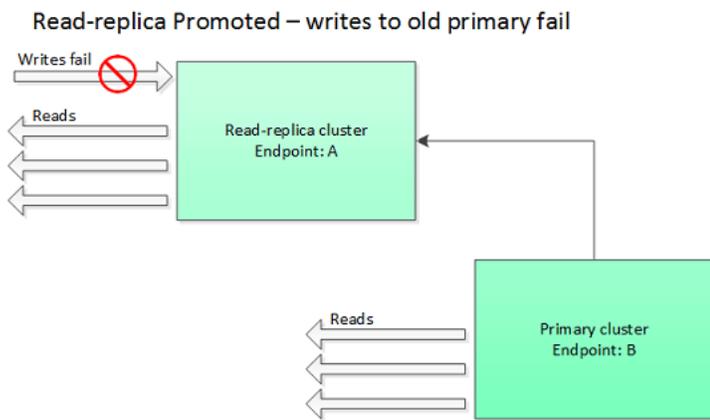
Como o `close-on-slave-write` funciona

O parâmetro `close-on-slave-write` é introduzido pelo Amazon ElastiCache para proporcionar mais controle sobre como o cluster responde quando um nó primário e um nó de réplica de leitura trocam de função devido à promoção de uma réplica de leitura para primário.

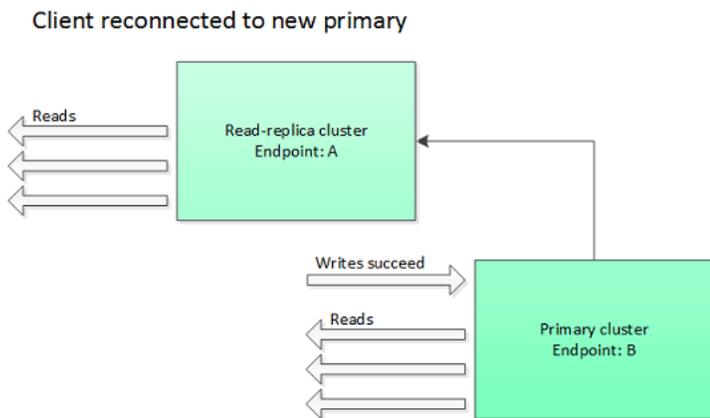
Before read-replica promotion



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. Este é o comportamento do Redis antes do ElastiCache introduzir `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. Sua lógica de aplicativo deve detectar a desconexão, verificar a tabela DNS e reconectar-se ao endpoint primário, que agora seria o endpoint B.



Quando você pode desativar o `close-on-replica-write`

Se desabilitar `close-on-replica-write` resulta na falha do cluster, por que você deseja desabilitar `close-on-replica-write`?

Conforme mencionado anteriormente, com `close-on-replica-write` habilitado, a qualquer momento que um cliente tentar gravar em uma réplica de leitura, a conexão do cliente com o cluster será encerrada. Estabelecer uma nova conexão com o nó leva tempo. Assim, desconectar e reconectar devido a uma solicitação de gravação para a réplica também afeta a latência das solicitações de leitura que são veiculadas pela mesma conexão. Esse efeito permanece em vigor até uma nova conexão ser estabelecida. Se seu aplicativo for especialmente pesado em termos de leitura ou muito sensível à latência, convém manter seus clientes conectados para evitar degradar o desempenho de leitura.

Parâmetros adicionados do Redis 2.8.22 (aprimorado)

Família de grupo de parâmetros: redis2.8

Para o Redis 2.8.22 não há parâmetros adicionais com suporte.

Important

- A partir do Redis versão 2.8.22, `repl-backlog-size` aplica-se ao cluster primário, bem como a clusters de réplica.
- A partir do Redis versão 2.8.22, o parâmetro `repl-timeout` não tem suporte. Se for alterado, o ElastiCache substituirá pelo padrão (60s), como fazemos com `appendonly`.

Os seguintes parâmetros não têm mais suporte.

- `appendonly`
- `appendfsync`
- `repl-timeout`

Parâmetros adicionados do Redis 2.8.21

Família de grupo de parâmetros: redis2.8

Para o Redis 2.8.21, não há parâmetros adicionais com suporte.

Parâmetros adicionados do Redis 2.8.19

Família de grupo de parâmetros: redis2.8

Para o Redis 2.8.19 não há parâmetros adicionais com suporte.

Parâmetros adicionados do Redis 2.8.6

Família de grupo de parâmetros: redis2.8

Para o Redis 2.8.6 há suporte para os parâmetros adicionais a seguir.

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 de <code>min-slaves-to-write</code> e, o primário deixará de aceitar gravações nesse ponto.</p> <p>Se esse parâmetro ou <code>min-slaves-to-write</code> for 0, o nó primário sempre aceitará solicitações de gravação, mesmo que não haja réplicas disponíveis.</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 não haja réplicas disponíveis.</p>
<code>notify-keyspace-events</code>	<p>Padrão: (uma string vazia)</p>	<p>Os tipos de eventos de espaço de chaves sobre os quais o Redis</p>

Nome	Detalhes	Descrição
	<p>Tipo: string</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>pode notificar os clientes. Cada tipo de evento é representado por uma única letra:</p> <ul style="list-style-type: none"> • K: eventos de espaço de chaves, publicados com um prefixo de <code>__keyspace@<db>__</code> • E: eventos-chave, publicados com um prefixo de <code>__keyevent@<db>__</code> • g — Comandos genéricos não específicos, como DEL, EXPIRE, RENAME, etc. • \$ — Comandos de string • l — Listar comandos • s — Definir comandos • h — Comandos de hash • z — Comandos de conjuntos classificados • x — Eventos expirados (eventos gerados sempre que uma chave expira) • e — Eventos removidos (eventos gerados quando uma chave é removida para <code>maxmemory</code>)

Nome	Detalhes	Descrição
		<ul style="list-style-type: none"><li data-bbox="1008 216 1446 279">• A — Um alias para g\$shzxe <p data-bbox="1008 384 1507 611">Você pode ter qualquer combinação desses tipos de eventos. Por exemplo, AKE significa que o Redis pode publicar notificações de todos os tipos de eventos.</p> <p data-bbox="1008 653 1474 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 1318"><p> Note</p><p>A partir do Redis 2.8.22, esse parâmetro se aplica ao cluster primário, bem como 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">• Transferência de dados em massa durante a sincronização, sob a perspectiva da réplica de leitura• Tempo limite do nó primário, sob a perspectiva da réplica• Tempo limite da réplica, sob a perspectiva do nó primário

Parâmetros do Redis 2.6.13

Família de grupo de parâmetros: redis2.6

O Redis 2.6.13 foi a primeira versão do Redis com suporte pelo ElastiCache. A tabela a seguir mostra os parâmetros do Redis 2.6.13 para os quais o ElastiCache oferece suporte.

Nome	Detalhes	Descrição
activerehashing	<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 sofre rehashing dez vezes por segundo. Cada operação de rehash consome 1 milissegundo de tempo da 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>Habilita ou desabilita o recurso de arquivo somente de acréscimo (AOF) do Redis. O AOF captura todos os comandos do Redis que alteram dados no cache e é usado para recuperação de determinadas falhas de nós.</p> <p>O valor padrão é no, o que significa que o AOF está desligado. Defina esse parâmetro como yes para habilitar o AOF.</p> <p>Para obter mais informações, consulte Atenuar falhas.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>O recurso Arquivos somente de acréscimo (AOF) não tem suporte para nós cache.t1.micro e cache.t2.*. Para</p> </div>

Nome	Detalhes	Descrição
		<p>nós desse tipo, o valor do parâmetro <code>appendonly</code> é ignorado.</p> <p> Note Para grupos de replicação Multi-AZ, o AOF não 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: <code>imediatamente</code></p>	<p>Quando <code>appendonly</code> é definido como <code>sim</code>, controla com que frequência o buffer de saída do AOF é gravado no disco:</p> <ul style="list-style-type: none"> • <code>no</code> — o buffer é liberado no disco, conforme necessário. • <code>everysec</code>: o buffer é liberado uma vez por segundo. Esse é o padrão. • <code>always</code> — o buffer é liberado sempre que os dados no cluster são modificados. • O <code>Appendfsync</code> não é compatível com as versões 2.8.22 e posteriores.
<code>client-output-buffer-limit-normal-hard-limit</code>	<p>Padrão: <code>0</code></p> <p>Tipo: <code>inteiro</code></p> <p>Modificável: <code>sim</code></p> <p>As alterações terão efeito: <code>imediatamente</code></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 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 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 por <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 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 tempo que esse número de segundos, o cliente será desconectado.
<code>client-output-buffer-limit-slave-hard-limit</code>	<p>Padrão: para valores, consulte Parâmetros específicos de node-type do Redis</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	Para réplicas de 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 Parâmetros específicos de node-type do Redis</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	Para réplicas de 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 por <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 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 tempo 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 de execução máximo para um script Lua, em milissegundos, antes que o ElastiCache tome medidas para interromper o script.</p> <p>Se <code>lua-time-limit</code> for excedido, todos os comandos do Redis retornarão um erro no formato <code>____-BUSY</code>. Como esse estado pode causar interferência com muitas operações essenciais do Redis, o ElastiCache primeiro emite um comando <code>SCRIPT KILL</code>. Se isso não tiver êxito, o ElastiCache reinicia o Redis de forma forçada.</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: 65000</p> <p>Tipo: inteiro</p> <p>Modificável: não</p>	
	<p>t3.small Padrão: 65000</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 Usar o Redis como um cache 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 de uso menos recente (LRU) e tempo de vida (TTL), esse parâmetro representa o tamanho de amostra das chaves para verificar. Por padrão, o Redis escolhe 3 chaves 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 nó Redis crescerá até consumir o <code>maxmemory</code> do nó (consulte Parâmetros específicos de node-type do Redis). Se isso ocorrer, o desempenho do nó provavelmente sofrerá devido a uma paginação de memória excessiva. Ao reservar memória, você pode separar uma parte da memória disponível para fins não Redis, a fim de ajudar a reduzir a quantidade de paginação.</p> <p>Este parâmetro é específico do ElastiCache e não faz parte da distribuição padrão do Redis.</p> <p>Para obter mais informações, consulte <code>reserved-memory-percent</code> e Gerenciamento da memória reservada.</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
<code>slave-all- ow-chaini ng</code>	<p>Padrão: não</p> <p>Tipo: string</p> <p>Modificável: não</p>	Determina se uma réplica de leitura no Redis pode ter réplicas de leitura próprias.
<code>slowlog-l og-slower -than</code>	<p>Padrão: 10000</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	O tempo máximo de execução, em microssegundos, para que os comandos sejam registrados pelo recurso Redis Slow Log.
<code>slowlog-m ax-len</code>	<p>Padrão: 128</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	O comprimento máximo do Redis Slow Log.
<code>tcp-keepa live</code>	<p>Padrão: 0</p> <p>Tipo: inteiro</p> <p>Modificável: sim</p> <p>As alterações terão efeito: imediatamente</p>	<p>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.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Alguns aspectos desse parâmetro mudaram na versão 3.2.4 do Redis. Consulte Parâmetros alterados no Redis 3.2.4 (aprimorado).</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"> • 0 – nunca desconectar um cliente ocioso. • 1-19 – valores inválidos. • ≥ 20 – o número de segundos que um nó espera antes de desconectar um cliente ocioso.
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 do Redis 2.6.13, um grupo de parâmetros padrão (`default.redis2.6`) será usado. Não é possível alterar os valores de parâmetros no grupo de parâmetros padrão. No entanto, você sempre pode criar um grupo de parâmetros personalizado e atribuí-lo ao seu cluster a qualquer momento.

Parâmetros específicos de node-type do Redis

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	client-output-buffer-limit-slave-hard-limit	client-output-buffer-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	client-output-buffer-limit-slave-hard-limit	client-output-buffer-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	client-output-buffer-limit-slave-hard-limit	client-output-buffer-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	client-output-buffer-limit-slave-hard-limit	client-output-buffer-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	client-output-buffer-limit-slave-hard-limit	client-output-buffer-limit-slave-soft-limit
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

Note

Todos os tipos de instâncias da geração atual são criados em uma Amazon Virtual Private Cloud VPC por padrão.

As instâncias T1 não oferecem suporte a Multi-AZ.

As instâncias de T1 e T2 não oferecem suporte ao AOF do Redis.

As variáveis de configuração do Redis `appendonly` e `appendfsync` não são compatíveis no Redis versão 2.8.22 e posterior.

Dimensionamento ElastiCache para Redis

Dimensionamento sem servidor ElastiCache

ElastiCache O Serverless acomoda automaticamente seu tráfego de carga de trabalho à medida que ele aumenta ou diminui. Para cada cache ElastiCache sem servidor, rastreia ElastiCache continuamente a utilização de recursos como CPU, memória e rede. Quando qualquer um desses recursos está restrito, o ElastiCache Serverless se expande adicionando um novo fragmento e redistribuindo os dados ao novo fragmento, sem nenhum tempo de inatividade para seu aplicativo. Você pode monitorar os recursos consumidos pelo seu cache CloudWatch monitorando a `BytesUsedForCache` métrica para armazenamento de dados em cache e `ElastiCacheProcessingUnits` (ECPU) para uso de computação.

Definição dos limites de escalabilidade para gerenciar custos

Você pode optar por configurar um uso máximo no armazenamento de dados em cache e na ECPU/segundo do cache para controlar os custos do cache. Isso vai garantir que o uso do cache jamais exceda o máximo configurado.

Se você definir uma escala máxima, a aplicação pode apresentar um desempenho de cache reduzido quando o cache atinge o máximo. Quando você define um máximo de armazenamento de dados em cache e seu armazenamento de dados em cache atinge o máximo, ElastiCache começará a despejar dados em seu cache que tenham um tempo de vida útil (TTL) definido, usando a lógica LRU. Se não houver dados que possam ser removidos, as solicitações de gravação de dados adicionais receberão uma mensagem de erro de falta de memória (OOM). Quando você define um máximo de ECPU/segundo e a utilização computacional de sua carga de trabalho excede esse valor, ElastiCache começará a limitar as solicitações do Redis.

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-escalamento com Serverless ElastiCache

ElastiCache Pré-escalamento sem servidor

Com o pré-escalamento, também chamado de pré-aquecimento, você pode definir limites mínimos suportados para seu cache. ElastiCache Você pode definir esses mínimos para unidades de ElastiCache 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-escalamento usando o ElastiCache console, a CLI ou a API. ElastiCache O Serverless atualiza as ECPUs disponíveis por segundo no cache em 60 minutos e envia uma notificação de evento quando a atualização do limite mínimo é concluída.

Como funciona o pré-escalamento

Quando o limite mínimo de ECPUs por segundo ou armazenamento de dados é atualizado por meio do console, da CLI ou da API, esse novo limite fica disponível em 1 hora. ElastiCache O Serverless

suporta 30K ECPUS/segundo em um cache vazio e até 90K ECPUS/seg ao usar o recurso Read from Replica. ElastiCache pode dobrar ECPUS/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 por segundo para o pico de ECPUs por segundo que você espera 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 as novas eCPUs mínimas por segundo ou o novo armazenamento mínimo. Isso ocorre mesmo se o aplicativo não estiver executando solicitações no cache ou se o uso do armazenamento de dados estiver abaixo do mínimo. Quando você reduz o limite mínimo da configuração atual, a atualização é imediata, então o ElastiCache Serverless começará a medir o novo limite mínimo imediatamente.

Note

- Ao definir um limite mínimo de uso, você é cobrado por esse limite, mesmo que seu uso real seja menor que o limite mínimo de uso. O uso de ECPU ou armazenamento de dados que exceda o limite mínimo de uso é cobrado de acordo com a taxa normal. Por exemplo, se você definir um limite mínimo de uso de 100.000 ECPUs por segundo, será cobrado pelo menos 1,224 USD por hora (usando os preços de ECPU em us-east-1), mesmo que seu uso seja inferior ao mínimo definido.
- ElastiCache O Serverless suporta a escala mínima solicitada em um nível agregado no cache. ElastiCache O Serverless também suporta um máximo de 30K ECPUS/segundo por slot (90K ECPUS/segundo ao usar Read from Replica usando conexões READONLY). Como prática recomendada, seu aplicativo deve garantir que a distribuição de chaves nos slots do Redis e o tráfego entre as chaves sejam os mais uniformes possíveis.

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 Ações 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 escalabilidade usando a CLI, use `modify-serverless-cache` a API.

Linux

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Remoção dos limites de escala usando a CLI

Para remover os limites de escala usando a CLI, defina o parâmetro 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 ElastiCache para clusters do Redis projetados pelo próprio usuário

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.

O conteúdo a seguir ajuda a encontrar o tópico correto para as ações de escalabilidade que você deseja executar.

Escalabilidade de clusters do Redis

Ação	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
Redução de escala	Remoção de nós de um cluster	Escalabilidade de clusters no Redis (modo cluster habilitado)
Aumento de escala	Adição de nós a um cluster	Refragmentação e rebalanceamento de fragmentos online para o Redis (modo cluster habilitado)
Alteração nos tipos de nó	Para um tipo de nó maior: <ul style="list-style-type: none"> Aumento de escala vertical de clusters de nó único do Redis (modo cluster desabilitado) 	Escalabilidade vertical online com modificação do tipo de nó

Ação	Redis (modo cluster desabilitado)	Redis (modo cluster habilitado)
	<p>Aumento de escala vertical de clusters do Redis com réplicas</p> <p>Para um tipo de nó menor:</p> <ul style="list-style-type: none"> • Redução de escala vertical de clusters do Redis de nó único • Redução de escala vertical de clusters do Redis com réplicas 	
Alteração do número de grupos de nós	Não há suporte para clusters do Redis (modo cluster desabilitado)	Escalabilidade de clusters no Redis (modo cluster habilitado)

Tópicos

- [Escalabilidade de clusters do Redis \(modo cluster desabilitado\)](#)
- [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#)

Escalabilidade de clusters do Redis (modo cluster desabilitado)

Os clusters do Redis (modo cluster desabilitado) podem ser um cluster de nó único com 0 fragmentos ou clusters de vários nós com 1 fragmento. Os clusters de nó único usam esse único nó para leituras e gravações. 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.

Sumário

- [Escalabilidade de clusters de nó único do Redis \(modo cluster desabilitado\)](#)
 - [Aumento de escala vertical de clusters de nó único do Redis \(modo cluster desabilitado\)](#)
 - [Aumento de escala vertical de clusters de nó único do Redis \(modo cluster desabilitado\) \(console\)](#)
 - [Aumento de escala vertical de clusters de cache do Redis de um único nó \(AWS CLI\)](#)
 - [Aumento de escala vertical de clusters de cache do Redis de um único nó \(API do ElastiCache\)](#)
 - [Redução de escala vertical de clusters do Redis de nó único](#)
 - [Redução de escala vertical de um cluster do Redis de nó único \(console\)](#)
 - [Redução de escala vertical de clusters de cache do Redis de nó único \(AWS CLI\)](#)
 - [Redução de escala vertical de clusters de cache do Redis de um único nó \(API do ElastiCache\)](#)
- [Escalabilidade de clusters do Redis \(modo cluster desativado\) com nós de réplica](#)
 - [Aumento de escala vertical de clusters do Redis com réplicas](#)
 - [Redução de escala vertical de clusters do Redis com réplicas](#)
 - [Aumento da capacidade de leitura](#)
 - [Redução da capacidade de leitura](#)

Escalabilidade de clusters de nó único do Redis (modo cluster desabilitado)

Nós do Redis (modo cluster desabilitado) devem ser suficientemente grandes para conter todos os dados do cache mais a sobrecarga do Redis. Para alterar a capacidade de dados do seu cluster do Redis (modo cluster desabilitado), você deve escalar verticalmente, expandindo para um tipo de nó maior para aumentar a capacidade de dados ou reduzindo para um tipo de nó menor para diminuir a capacidade de dados.

O processo de aumento de escala vertical do ElastiCache for Redis foi projetado para fazer o melhor esforço no sentido de reter seus dados existentes e requer uma replicação do Redis bem-sucedida. Para clusters do Redis (modo de cluster desabilitado), recomendamos que seja disponibilizada memória suficiente para o Redis.

Não é possível particionar seus dados em vários clusters Redis (modo cluster desabilitado). No entanto, se você só precisa aumentar ou diminuir a capacidade de leitura do seu cluster, pode criar um cluster do Redis (modo cluster desabilitado) com nós de réplica e adicionar ou remover réplicas de leitura. Para criar um cluster do Redis (modo cluster desabilitado) com nós de réplica usando seu cluster de cache Redis de um único nó como cluster primário, consulte [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#).

Depois de criar o cluster com réplicas, você pode aumentar a capacidade de leitura adicionando réplicas de leitura. Mais tarde, se precisar, você pode reduzir a capacidade de leitura removendo réplicas de leitura. Para ter mais informações, consulte [Aumento da capacidade de leitura](#) ou [Redução da capacidade de leitura](#).

Além de poderem escalonar a capacidade de leitura, clusters do Redis (modo cluster desabilitado) com réplicas fornecem outras vantagens comerciais. Para obter mais informações, consulte [Alta disponibilidade com o uso de grupos de replicação](#).

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Redis, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster. Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

Tópicos

- [Aumento de escala vertical de clusters de nó único do Redis \(modo cluster desabilitado\)](#)
- [Redução de escala vertical de clusters do Redis de nó único](#)

Aumento de escala vertical de clusters de nó único do Redis (modo cluster desabilitado)

Quando você aumenta a escala vertical de um cluster Redis de um único nó, o ElastiCache realiza o processo a seguir, independentemente de você usar o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

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 perceberá uma breve interrupção (alguns segundos) das leituras e gravações no nó primário enquanto a entrada do DNS for atualizada.
4. O ElastiCache exclui o cluster de cache antigo. Você perceberá uma breve interrupção (alguns segundos) das leituras e gravações no nó antigo porque as conexões ao nó antigo serão desconectadas.

Note

Para clusters que executam o tipo de nó r6gd, você só pode escalar para tamanhos de nó dentro da família de nós r6gd.

Conforme mostrado na tabela a seguir, sua operação de expansão do Redis será bloqueada se você tiver uma atualização do mecanismo programada para a próxima janela de manutenção. Para obter mais informações sobre janelas de manutenção, consulte [Gerenciamento da manutenção](#).

Operações Redis bloqueadas

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.

- Planeje sua operação de expansão do Redis para a próxima janela de manutenção, desmarcando a caixa de seleção Apply immediately (CLI use: `--no-apply-immediately`, API use: `ApplyImmediately=false`).
- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de expansão do Redis.
- Adicione a atualização do mecanismo Redis a essa modificação de cluster de cache com a caixa de seleção Apply Immediately marcada (CLI use: `--apply-immediately`, API use: `ApplyImmediately=true`). Isso desbloqueia sua operação de expansão, fazendo com que a atualização do mecanismo seja realizada imediatamente.

Você pode aumentar a escala vertical de um cluster do Redis (modo cluster desabilitado) de um único nó usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache.

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Redis, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster. Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

Aumento de escala vertical de clusters de nó único do Redis (modo cluster desabilitado) (console)

O procedimento a seguir descreve como expandir um cluster do Redis de um nó único usando o Console de Gerenciamento do ElastiCache. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala vertical de um cluster Redis de nó único (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).
3. Na lista de clusters, escolha o cluster que você deseja expandir (ele deve estar executando o mecanismo Redis, e não o mecanismo Clustered Redis).
4. Escolha Modify (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 Modify (Modificar).

Se você escolheu Apply immediately na etapa anterior, o status do cluster mudará para `modifying`. Quando o status mudar para `available`, a modificação estará completa, e você poderá começar a usar o novo cluster.

Aumento de escala vertical de clusters de cache do Redis de um único nó (AWS CLI)

O procedimento a seguir descreve como expandir um cluster de cache Redis de um único nó usando a AWS CLI. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala verticalmente de um cluster de cache Redis de um único nó (AWS CLI)

1. Determine os tipos de nó para os quais você pode expandir executando o comando AWS CLI da `list-allowed-node-type-modifications` com o seguinte parâmetro.
 - `--cache-cluster-id`

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ],  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small"  
  ],  
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu cluster de cache existente, especificando o cluster de cache a ser expandido e o novo tipo de nó maior, usando o comando AWS CLI da `modify-cache-cluster` e os seguintes parâmetros.

- `--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-x1 \  
  --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-x1 ^  
  --apply-immediately
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "CacheCluster": {  
    "Engine": "redis",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.redis6.x",
```

```

        "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
        "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
}
}

```

Para obter mais informações, consulte [modify-cache-cluster](#) na AWS CLI Referência.

3. Se você usou `--apply-immediately`, verifique o status do novo cluster de cache usando o comando AWS CLI da `describe-cache-clusters` com o seguinte parâmetro. Quando o status mudar para `available`, você poderá começar a usar o novo cluster de cache maior.
 - `--cache-cluster-id`: o nome do cluster de cache do Redis 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 AWS CLI Referência.

Aumento de escala vertical de clusters de cache do Redis de um único nó (API do ElastiCache)

O procedimento a seguir descreve como aumentar a escala vertical de um cluster de cache Redis de um único nó usando a API do ElastiCache. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala vertical de um cluster de cache do Redis de um único nó (API do ElastiCache)

1. Determine os tipos de nó para os quais você pode aumentar a escala vertical executando a ação `ListAllowedNodeTypeModifications` da API do ElastiCache com o seguinte parâmetro.
 - `CacheClusterId` - O nome do cluster de cache Redis de nó único que você deseja expandir.

```
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 Referência da API do Amazon ElastiCache.

2. Modifique seu cluster de cache existente, especificando o cluster de cache a ter aumento de escala vertical e o novo tipo de nó maior, usando a ação `ModifyCacheCluster` da API do ElastiCache e os seguintes parâmetros.
 - `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 ação `ListAllowedNodeTypeModifications` na etapa 1.
 - `CacheParameterGroupName` – [Opcional] Use esse parâmetro se estiver usando `reserved-memory` para gerenciar a memória reservada do cluster. Especifique um `parameter group` de cache personalizado que reserve a quantidade correta de memória para

o seu novo tipo de nó. Se estiver usando `reserved-memory-percent`, você poderá omitir esse parâmetro.

- `ApplyImmediately`: defina como `true` para fazer com que o processo de expansão seja executado imediatamente. Para adiar o processo de expansão para a próxima janela de manutenção do cluster, use `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &CacheClusterId=MyRedisCacheCluster  
  &CacheNodeType=cache.m3.xlarge  
  &CacheParameterGroupName redis32-m2-x1  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyCacheCluster](#) na Referência da API do Amazon ElastiCache.

3. Se você usou `ApplyImmediately=true`, verifique o status do novo cluster de cache usando a ação `DescribeCacheClusters` da API do ElastiCache com o seguinte parâmetro. Quando o status mudar para `available`, você poderá começar a usar o novo cluster de cache maior.
 - `CacheClusterId`: o nome do cluster de cache do Redis 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 Referência da API do Amazon ElastiCache.

Redução de escala vertical de clusters do Redis de nó único

As seções a seguir demonstram como reduzir um cluster Redis de nó único para um tipo de nó menor. Garantir que o novo tipo de nó menor seja suficientemente grande para acomodar todos os dados e a sobrecarga do Redis é importante para o sucesso a longo prazo do seu novo cluster Redis. Para obter mais informações, consulte [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#).

Note

Para clusters que executam o tipo de nó `r6gd`, você só pode escalar para tamanhos de nó dentro da família de nós `r6gd`.

Tópicos

- [Redução de escala vertical de um cluster do Redis de nó único \(console\)](#)
- [Redução de escala vertical de clusters de cache do Redis de nó único \(AWS CLI\)](#)
- [Redução de escala vertical de clusters de cache do Redis de um único nó \(API do ElastiCache\)](#)

Redução de escala vertical de um cluster do Redis de nó único (console)

O procedimento a seguir demonstra passo a passo o processo de reduzir a escala vertical do seu cluster Redis de um único nó para um tipo de nó menor usando o console do ElastiCache.

Important

Se o seu parameter group usar `reserved-memory` para reservar a memória para a sobrecarga do Redis, antes de iniciar a escalabilidade, certifique-se de ter um parameter group personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um parameter group personalizado para que ele use `reserved-memory-percent` e usar esse parameter group para seu novo cluster. Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

Para reduzir a escala vertical do seu cluster do 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 o seu parameter group usar `reserved-memory` para reservar memória para a sobrecarga do Redis, certifique-se de ter um parameter group personalizado para reservar a quantidade correta de memória para o seu novo tipo de nó.

Alternativamente, você pode modificar seu parameter group personalizado para usar `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

3. Faça login no AWS Management Console e abra o console do ElastiCache 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 Redis, e não o mecanismo Clustered Redis.
5. Escolha Modify (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 Modify (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.

Redução de escala vertical de clusters de cache do Redis de nó único (AWS CLI)

O procedimento a seguir descreve como reduzir um cluster de cache Redis de um único nó usando a AWS CLI.

Para reduzir a escala verticalmente de um cluster de cache Redis de um único nó (AWS CLI)

1. Determine os tipos de nó que você pode reduzir executando o comando `list-allowed-node-type-modifications` da AWS CLI com o seguinte parâmetro.

- `--cache-cluster-id`

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small",  
  ],  
}
```

```
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique o cluster de cache existente, especificando o cluster de cache a ser reduzido e o novo tipo de nó menor, usando o comando `modify-cache-cluster` da AWS CLI 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 semelhante a esta (formato JSON).

```
{
  "CacheCluster": {
    "Engine": "redis",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.redis6.x",
      "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
      "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
  }
}
```

Para obter mais informações, consulte [modify-cache-cluster](#) na AWS CLI Referência.

3. Se você usou `--apply-immediately`, verifique o status do novo cluster de cache usando o comando AWS CLI da `describe-cache-clusters` com o seguinte parâmetro. Quando o status mudar para `available`, você poderá começar a usar o novo cluster de cache maior.

- `--cache-cache cluster-id`: o nome do cluster de cache do Redis 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 AWS CLI Referência.

Redução de escala vertical de clusters de cache do Redis de um único nó (API do ElastiCache)

O procedimento a seguir descreve como reduzir a escala vertical de um cluster de cache do Redis de um único nó usando a API do ElastiCache.

Para reduzir a escala vertical de um cluster de cache do Redis de um único nó (API do ElastiCache)

1. Determine os tipos de nó para os quais você pode reduzir a escala vertical executando a ação `ListAllowedNodeTypeModifications` da API do ElastiCache com o seguinte parâmetro.
 - `CacheClusterId`: o nome do cluster de cache do Redis de nó único que você deseja reduzir a escala vertical.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ListAllowedNodeTypeModifications](#) na Referência da API do Amazon ElastiCache.

2. Modifique seu cluster de cache existente, especificando o cluster de cache a ter aumento de escala vertical e o novo tipo de nó maior, usando a ação `ModifyCacheCluster` da API do ElastiCache e os seguintes parâmetros.
 - `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 nós retornados pela ação `ListAllowedNodeTypeModifications` na etapa 1.
- **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 Referência da API do Amazon ElastiCache.

3. Se você usou `ApplyImmediately=true`, verifique o status do novo cluster de cache usando a ação `DescribeCacheClusters` da API do ElastiCache com o seguinte parâmetro. Quando o status mudar para `available` (disponível), você poderá começar a usar o novo cluster de cache menor.
 - **CacheClusterId**: o nome do cluster de cache do Redis 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 Referência da API do Amazon ElastiCache.

Escalabilidade de clusters do Redis (modo cluster desativado) com nós de réplica

Um cluster Redis com nós de réplica (chamado de grupo de replicação na API/CLI) fornece alta disponibilidade por meio de uma replicação que possui o recurso Multi-AZ com failover automático habilitado. Um cluster com nós de réplica é um conjunto lógico de até seis clusters Redis em que um nó, o primário, pode atender a solicitações de leitura e gravação. Todos os outros nós do cluster são réplicas somente leitura do primário. Os dados escritos no primário são replicados de forma assíncrona para todas as réplicas de leitura no cluster. Como o Redis (modo cluster desabilitado) não oferece suporte ao particionamento dos seus dados em vários clusters, cada nó em um grupo de replicação do Redis (modo cluster desabilitado) contém todo o conjunto de dados do cache. Clusters do Redis (modo cluster habilitado) oferecem suporte para o particionamento dos seus dados até 500 fragmentos.

Para alterar a capacidade de dados do seu cluster, você deve expandi-lo para um tipo de nó maior ou reduzi-lo para um tipo de nó menor.

Para alterar a capacidade de leitura do seu cluster, adicione mais réplicas de leitura, até um máximo de 5, ou remova réplicas de leitura.

O processo de aumento de escala vertical do ElastiCache foi projetado para fazer o melhor esforço no sentido de reter seus dados existentes e requer uma replicação Redis bem-sucedida. Para clusters Redis com réplicas, recomendamos que uma memória suficiente seja disponibilizada para o Redis.

Tópicos relacionados

- [Alta disponibilidade com o uso de grupos de replicação](#)
- [Replicação: Redis \(modo cluster desabilitado\) versus Redis \(modo cluster habilitado\)](#)
- [Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ](#)
- [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#)

Tópicos

- [Aumento de escala vertical de clusters do Redis com réplicas](#)
- [Redução de escala vertical de clusters do Redis com réplicas](#)
- [Aumento da capacidade de leitura](#)
- [Redução da capacidade de leitura](#)

Aumento de escala vertical de clusters do Redis com réplicas

O Amazon ElastiCache fornece suporte de console, CLI e API para o aumento de escala vertical do seu grupo de replicação do Redis (modo cluster desabilitado).

Quando o processo de aumento de escala vertical é iniciado, o ElastiCache faz o seguinte:

1. Executa um grupo de replicação usando o novo tipo de nó.
2. Copia todos os dados do nó primário atual para o novo nó primário.
3. Sincroniza as novas réplicas de leitura com o novo nó primário.
4. Atualiza as entradas do DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Redis 5.0.5 e acima, você pode dimensionar clusters habilitados para failover automático enquanto o cluster permanece online e atende às solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.
5. Exclui os nós antigos (CLI/API: grupo de replicação). Você perceberá uma breve interrupção (alguns segundos) nas leituras e gravações dos nós antigos porque as conexões aos nós antigos serão desconectadas.

O tempo necessário para esse processo depende do tipo de nó e da quantidade de dados no seu cluster.

Conforme mostrado na tabela a seguir, sua operação de expansão do Redis será bloqueada se você tiver uma atualização do mecanismo programada para a próxima janela de manutenção do cluster.

Operações Redis bloqueadas

Operações pendentes	Operações bloqueadas
Amplie a sua capacidade	Atualização imediata do mecanismo
Atualização do mecanismo	Expansão imediata
Expansão e atualização do mecanismo	Expansão imediata
	Atualização imediata do mecanismo

Se você tiver uma operação pendente que está bloqueando você, será possível tomar uma destas providências.

- Planeje sua operação de expansão do Redis para a próxima janela de manutenção, desmarcando a caixa de seleção `Apply immediately` (CLI use: `--no-apply-immediately`, API use: `ApplyImmediately=false`).
- Aguarde até sua próxima janela de manutenção (ou depois) para realizar sua operação de expansão do Redis.
- Adicione a atualização do mecanismo Redis a essa modificação de cluster de cache com a caixa de seleção `Apply Immediately` marcada (CLI use: `--apply-immediately`, API use: `ApplyImmediately=true`). Isso desbloqueia sua operação de expansão, fazendo com que a atualização do mecanismo seja realizada imediatamente.

As seções a seguir descrevem como aumentar a escala vertical do seu cluster do Redis com réplicas usando o console do ElastiCache, a AWS CLI e a API do ElastiCache.

Important

Se o seu `parameter group` usar `reserved-memory` para reservar a memória para a sobrecarga do Redis, antes de iniciar a escalabilidade, certifique-se de ter um `parameter group` personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um `parameter group` personalizado para que ele use `reserved-memory-percent` e usar esse `parameter group` para seu novo cluster. Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

Aumento de escala vertical de um cluster do 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 console do ElastiCache. Durante esse processo, poderá haver uma breve interrupção de leituras e gravações de outras versões do nó primário enquanto a entrada de DNS é atualizada. Talvez você observe menos de 1 segundo de tempo de inatividade para os nós executados nas versões 5.0.5 e posteriores e alguns segundos para as versões mais antigas.

Para aumentar a escala vertical de um cluster do Redis com réplicas (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Redis).
3. Na lista de clusters, escolha o cluster que você deseja expandir. Esse cluster deve estar executando o mecanismo Redis, e não o mecanismo Clustered Redis.
4. Escolha Modify (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 Modify (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.

Aumento de escala vertical de um grupo de replicação do 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, o ElastiCache for Redis atualiza as entradas do DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Redis 5.0.5 e acima, você pode dimensionar clusters habilitados para failover automático enquanto o cluster permanece online e atende às solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.

O tempo necessário para 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 aumentar a escala verticalmente de um grupo de replicação Redis (AWS CLI)

1. Determine para quais tipos de nó você pode expandir executando o comando AWS CLI da `list-allowed-node-type-modifications` com o seguinte parâmetro.
 - `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-repl-group
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-repl-group
```

O resultado dessa operação é semelhante a este (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI

Referência.

2. Expanda seu grupo de replicação atual para o novo tipo de nó usando o comando AWS CLI da `modify-replication-group` com os seguintes parâmetros.
 - `--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 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 a operação de expansão para a próxima janela de manutenção, use `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-repl-group \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-repl-group ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

O resultado desse comando é semelhante a este (formato JSON).

```
{  
  "ReplicationGroup": {  
    "Status": "available",
```

```
"Description": "Some description",
"NodeGroups": [{
  "Status": "available",
  "NodeGroupMembers": [{
    "CurrentRole": "primary",
    "PreferredAvailabilityZone": "us-west-2b",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Port": 6379,
      "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
    },
    "CacheClusterId": "my-repl-group-001"
  },
  {
    "CurrentRole": "replica",
    "PreferredAvailabilityZone": "us-west-2c",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Port": 6379,
      "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
    },
    "CacheClusterId": "my-repl-group-002"
  }
  ],
  "NodeGroupId": "0001",
  "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
  }
}],
"ReplicationGroupId": "my-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "disabled",
"SnapshotWindow": "12:00-13:00",
"SnapshottingClusterId": "my-repl-group-002",
"MemberClusters": [
  "my-repl-group-001",
  "my-repl-group-002"
],
"PendingModifiedValues": {}
}
}
```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou o parâmetro `--apply-immediately`, monitore o status do grupo de replicação usando o comando AWS CLI da `describe-replication-group` com o seguinte parâmetro. Enquanto o status ainda for `modifying` (em modificação), você poderá observar menos de 1 segundo de tempo de inatividade para os nós executados nas versões 5.0.6 e posteriores, e uma breve interrupção de leituras e gravações para as versões mais antigas do nó primário enquanto a entrada do DNS é atualizada.
 - `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache describe-replication-groups \  
  --replication-group-id my-replication-group
```

Para Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id my-replication-group
```

Para obter mais informações, consulte [describe-replication-groups](#) na AWS CLI Referência.

Aumento de escala vertical de um grupo de replicação do Redis (API do ElastiCache)

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó maior usando a API do ElastiCache. Para o Redis 5.0.5 e acima, você pode dimensionar clusters habilitados para failover automático enquanto o cluster permanece online e atende às solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.

O tempo necessário para 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 aumentar a escala vertical de um grupo de replicação do Redis (API do ElastiCache)

1. Determine quais tipos de nós você pode aumentar a escala vertical usando a ação `ListAllowedNodeTypeModifications` da API do ElastiCache com o seguinte parâmetro.
 - `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 Referência da API do Amazon ElastiCache.

2. Aumente a escala vertical do seu grupo de replicação atual para o novo tipo de nó usando a ação `ModifyRedplicationGroup` da API do ElastiCache e com os seguintes parâmetros.
 - `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 ação `ListAllowedNodeTypeModifications` na etapa 1.
 - `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 Referência da API do Amazon ElastiCache.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a ação `DescribeReplicationGroups` da API do ElastiCache 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 Referência da API do Amazon ElastiCache.

Redução de escala vertical de clusters do Redis com réplicas

As seções a seguir demonstram passo a passo o processo de reduzir a escala vertical de um cluster de cache Redis (modo cluster desabilitado) com nós de réplica para um tipo de nó menor. Garantir que o novo tipo de nó menor seja suficientemente grande para acomodar todos os dados e a sobrecarga é muito importante para o sucesso. Para obter mais informações, consulte [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#).

Note

Para clusters que executam o tipo de nó `r6gd`, você só pode escalar para tamanhos de nó dentro da família de nós `r6gd`.

Important

Se o seu `parameter group` usar `reserved-memory` para reservar a memória para a sobrecarga do Redis, antes de iniciar a escalabilidade, certifique-se de ter um `parameter group` personalizado que reserve a quantidade correta de memória para o seu novo tipo de nó. Como alternativa, você pode modificar um `parameter group` personalizado para que ele use `reserved-memory-percent` e usar esse `parameter group` para seu novo cluster. Se você estiver usando `reserved-memory-percent`, não será necessário fazer isso. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

Redução de escala vertical de um cluster do Redis com réplicas (console)

O processo a seguir escala seu cluster Redis com nós de réplica para um tipo de nó menor usando o console do ElastiCache.

Para reduzir a escala vertical um cluster do 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 o seu `parameter group` usar `reserved-memory` para reservar memória para a sobrecarga do Redis, certifique-se de ter um `parameter group` personalizado para reservar a quantidade correta de memória para o seu novo tipo de nó.

Alternativamente, você pode modificar seu parameter group personalizado para usar `reserved-memory-percent`. Para obter mais informações, consulte [Gerenciamento da memória reservada](#).

3. Faça login no AWS Management Console e abra o console do ElastiCache 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 Redis, e não o mecanismo Clustered Redis.
5. Escolha Modify (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 Modify (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.

Redução de escala vertical de um grupo de replicação do 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, o ElastiCache for Redis atualiza as entradas do DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Redis 5.0.5 e acima, você pode dimensionar clusters habilitados para failover automático enquanto o cluster permanece online e atende às solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.

No entanto, as leituras dos clusters de 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 a escala verticalmente de um grupo de replicação do Redis (AWS CLI)

1. Determine para quais tipos de nó você pode reduzir executando o comando `list-allowed-node-type-modifications` da AWS CLI com o parâmetro a seguir.
 - `--replication-group-id` - o nome do grupo de replicação. Use esse parâmetro para descrever um grupo de replicação específico em vez de todos os grupos de replicação.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-repl-group
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-repl-group
```

O resultado dessa operação é semelhante a este (formato JSON).

```
{  
  "ScaleDownModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI

Referência.

2. Expanda seu grupo de replicação atual para o novo tipo de nó usando o comando AWS CLI da `modify-replication-group` com os seguintes parâmetros.
 - `--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 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 a operação de expansão para a próxima janela de manutenção, use `--no-apply-immediately`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-repl-group \  
  --cache-node-type cache.t2.small \  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-repl-group ^  
  --cache-node-type cache.t2.small ^  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

O resultado desse comando é semelhante a este (formato JSON).

```
{"ReplicationGroup": {  
  "Status": "available",  
  "Description": "Some description",
```

```

"NodeGroups": [
  {
    "Status": "available",
    "NodeGroupMembers": [
      {
        "CurrentRole": "primary",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "my-repl-group-001"
      },
      {
        "CurrentRole": "replica",
        "PreferredAvailabilityZone": "us-west-2c",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
        },
        "CacheClusterId": "my-repl-group-002"
      }
    ],
    "NodeGroupId": "0001",
    "PrimaryEndpoint": {
      "Port": 6379,
      "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
    }
  }
],
"ReplicationGroupId": "my-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "disabled",
"SnapshotWindow": "12:00-13:00",
"SnapshottingClusterId": "my-repl-group-002",
"MemberClusters": [
  "my-repl-group-001",
  "my-repl-group-002",
],

```

```
    "PendingModifiedValues": {}  
  }  
}
```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou o parâmetro `--apply-immediately`, monitore o status do grupo de replicação usando o comando AWS CLI `describe-replication-group` com o seguinte parâmetro. 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.

Redução de escala vertical de um grupo de replicação do Redis (API do ElastiCache)

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando a API do ElastiCache. Durante esse processo, o ElastiCache for Redis atualiza as entradas do DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Redis 5.0.5 e acima, você pode dimensionar clusters habilitados para failover automático enquanto o cluster permanece online e atende às solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada. No entanto, as leituras dos clusters de 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 a escala vertical de um grupo de replicação do Redis (API do ElastiCache)

1. Determine quais tipos de nós você pode reduzir a escala vertical usando a ação `ListAllowedNodeTypeModifications` da API do ElastiCache com o seguinte parâmetro.
 - `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 Referência da API do Amazon ElastiCache.

2. Aumente a escala vertical do seu grupo de replicação atual para o novo tipo de nó usando a ação `ModifyRedplicationGroup` da API do ElastiCache e com os seguintes parâmetros.
 - `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 ação `ListAllowedNodeTypeModifications` na etapa 1.
 - `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 Referência da API do Amazon ElastiCache.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a ação `DescribeReplicationGroups` da API do ElastiCache 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 Referência da API do Amazon ElastiCache.

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 Redis.

Você pode escalar a capacidade de leitura do cluster do Redis usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Para obter mais informações, consulte [Adicionando uma réplica de leitura, para grupos de replicação do Redis \(Modo cluster desabilitado\)](#).

Redução da capacidade de leitura

Para diminuir a capacidade de leitura, exclua uma ou mais réplicas de leitura do seu cluster Redis com réplicas (chamado de grupo de replicação na API/CLI). Se o cluster for habilitado para Multi-AZ com failover automático, não será possível excluir a última réplica de leitura sem primeiro desabilitar o Multi-AZ. Para obter mais informações, consulte [Modificação de um grupo de replicação](#).

Para obter mais informações, consulte [Exclusão de uma réplica de leitura, para grupos de replicação do Redis \(modo cluster desabilitado\)](#).

Escalabilidade de clusters no Redis (modo cluster habilitado)

À medida que a demanda nos clusters muda, você pode optar por aumentar a performance ou reduzir os custos alterando o número de fragmentos no seu cluster do Redis (modo cluster habilitado). Recomendamos o uso da escalabilidade horizontal online para esse ajuste, pois permite que o seu cluster continue a atender às solicitações durante o processo de escalabilidade.

As condições sob as quais você pode decidir redimensionar seu cluster incluem o seguinte:

- **Uso intenso de memória:**

Se os nós no cluster estão sob uso intenso da memória, você pode optar por aumentar a escala e ter mais recursos para melhor armazenar dados e atender a solicitações.

Você pode determinar se os nós estão sob uso intenso da memória monitorando as seguintes métricas: `FreeableMemory`, `SwapUsage` e `BytesUseForCache`.

- **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 throughput monitorando as seguintes métricas: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections` 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 a escala pode ser reduzida na horizontal com segurança usando as seguintes métricas: `FreeableMemory`, `SwapUsage`, `BytesUseForCache`, `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections` e `NewConnections`.

Impacto da escalabilidade no desempenho

Quando você altera a escala usando o processo offline, seu cluster fica offline para uma parte significativa do processo e, por conseguinte, não é capaz de atender a solicitações. Quando você altera a escala usando o método online, como a escalabilidade é uma operação com uso intensivo de computação, há queda no desempenho, mas ainda assim seu cluster continua atendendo a

solicitações durante a operação de escalabilidade. O quanto o desempenho é afetado depende do seu uso normal da CPU e dos seus dados.

Existem duas maneiras de escalar seu cluster do Redis (modo cluster desabilitado): com escalabilidade horizontal e vertical.

- A escalabilidade horizontal permite alterar o número de grupos de nós (estilhaços) no grupo de replicação adicionando ou removendo grupos de nós (estilhaços). O processo de 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 estiver reduzindo o tamanho e a capacidade de memória do cluster, reduzindo ou expandindo, garanta que a nova configuração tenha memória suficiente para seus dados e a sobrecarga do Redis.

Para obter mais informações, consulte [Select cache node size](#).

Sumário

- [Refragmentação e rebalanceamento de fragmentos offline para o Redis \(modo cluster habilitado\)](#)
- [Refragmentação e rebalanceamento de fragmentos online para o Redis \(modo cluster habilitado\)](#)
 - [Adição de fragmentos com refragmentação online](#)
 - [Remoção de fragmentos com refragmentação online](#)
 - [Remoção de fragmentos \(console\)](#)
 - [Remoção de fragmentos \(AWS CLI\)](#)
 - [Removendo fragmentos \(ElastiCacheAPI\)](#)
 - [Rebalanceamento de fragmentos online](#)
 - [Rebalanceamento de fragmentos online \(Console\)](#)
 - [Rebalanceamento de fragmentos online \(AWS CLI\)](#)
 - [Rebalanceamento de fragmentos on-line \(API\) ElastiCache](#)
- [Escalabilidade vertical online com modificação do tipo de nó](#)

- [Aumento de escala vertical on-line](#)
 - [Aumento de escala vertical de clusters de cache do Redis \(console\)](#)
 - [Aumento de escala vertical de clusters de cache do Redis \(AWS CLI\)](#)
 - [Aumento de escala vertical de clusters de cache do Redis \(API do ElastiCache\)](#)
- [Redução de escala vertical online](#)
 - [Redução de escala vertical de clusters de cache do Redis \(console\)](#)
 - [Redução de escala vertical de clusters de cache do Redis \(AWS CLI\)](#)
 - [Redução de escala vertical de clusters de cache do Redis \(API do ElastiCache\)](#)

Refragmentação e rebalanceamento de fragmentos offline para o Redis (modo cluster habilitado)

A principal vantagem de obter a reconfiguração de estilhaços offline é que você pode fazer mais do que simplesmente adicionar ou remover estilhaços do seu grupo de replicação. Quando você faz o reestilhaçamento offline, além de alterar o número de estilhaços no seu grupo de replicação, é possível fazer o seguinte:

 Note

O reestilhaçamento offline não é compatível com clusters do Redis que tenham a classificação de dados em níveis ativada. Para obter mais informações, consulte [Classificação de dados em níveis](#).

- 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 estilhaços offline é que o cluster fica offline começando com a parte de restauração do processo e continua até você atualizar os endpoints no aplicativo. O tempo em que o cluster fica offline depende da quantidade de dados no seu cluster.

Para reconfigurar seus fragmentos offline no cluster do Redis (modo cluster habilitado)

1. Crie um backup manual do seu cluster Redis 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 [Localização de endpoints de conexão](#).

Refragmentação e rebalanceamento de fragmentos online para o Redis (modo cluster habilitado)

Ao usar a refragmentação on-line e o rebalanceamento de fragmentos com a versão 3.2.10 ou mais recente do ElastiCache Amazon for Redis, você pode escalar ElastiCache seu for Redis (modo de cluster ativado) dinamicamente sem tempo de inatividade. Essa abordagem significa que seu cluster pode continuar atendendo a solicitações mesmo durante a escalabilidade ou o rebalanceamento.

Você pode fazer o seguinte:

- Aumentar a escala horizontalmente: aumente a capacidade de leitura e gravação adicionando fragmentos (grupos de nós) ao seu cluster do Redis (modo cluster habilitado) (grupo de replicação).

Se você adicionar um ou mais fragmentos ao grupo de replicação, o número de nós em cada novo fragmento será o mesmo que o número de nós no menor dos fragmentos existentes.

- Reduzir a escala horizontalmente: reduza de capacidade de leitura e gravação e, por conseguinte, os custos, removendo fragmentos do cluster do Redis (modo cluster habilitado).
- Rebalancear — Mova os espaços de chave entre os fragmentos em seu cluster ElastiCache for Redis (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 ElastiCache refragmentação e rebalanceamento on-line do Redis:

- Esses processos exigem a versão 3.2.10 do mecanismo do Redis ou mais recente. Para obter informações sobre como atualizar sua versão de mecanismo, consulte [Versões de mecanismos e atualização](#).

- Há limitações em relação a slots ou espaços de chave e itens grandes:

Se qualquer uma das chaves em um fragmento contiver um item grande, essa chave não será migrada para um novo fragmento durante o aumento da escala ou o rebalanceamento. Essa funcionalidade pode resultar em fragmentos desbalanceados.

Se qualquer uma das chaves em um fragmento contiver um item grande (itens maiores do que 256 MB após a serialização), o fragmento não será excluído na redução da escala. Essa funcionalidade pode resultar na não exclusão de alguns fragmentos.

- Ao aumentar a escala, o número de nós em novos fragmentos fica igual ao número de nós no menor dos fragmentos existentes.
- No aumento de escala, qualquer tag comum a todos os fragmentos existentes é copiada para os novos fragmentos.
- Ao escalar um cluster de armazenamento de dados global, não ElastiCache replicará automaticamente as funções de um dos nós existentes para o (s) novo (s) nó (s). Recomendamos carregar suas funções nos novos fragmentos depois de aumentar a escala horizontalmente do seu cluster para que todos os fragmentos tenham as mesmas funções.

Note

No ElastiCache Redis, 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 do Redis](#), recomendamos carregar todas as suas funções em todos os fragmentos antes de escalar ElastiCache para que seu cluster do Redis não tenha definições de função diferentes em fragmentos diferentes.

Para ter mais informações, consulte [Redimensionamento de cluster on-line](#).

Você pode escalar ou reequilibrar horizontalmente seus clusters ElastiCache for Redis (modo de cluster ativado) usando a AWS Management Console, a e a API AWS CLI. ElastiCache

Adição de fragmentos com refragmentação online

Você pode adicionar fragmentos ao seu cluster Redis (modo de cluster ativado) usando a API AWS Management Console AWS CLI, ou ElastiCache. Quando você adiciona fragmentos a um cluster do Redis (modo cluster habilitado), qualquer tag nos fragmentos existentes é copiada para os novos fragmentos.

Adição de fragmentos (console)

Você pode usar o AWS Management Console para adicionar um ou mais fragmentos ao seu cluster Redis (modo de cluster ativado). O procedimento a seguir descreve o processo.

Como adicionar fragmentos ao seu cluster do Redis (modo cluster habilitado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).
3. Localize e escolha o nome, não a caixa à esquerda do nome do cluster, do cluster do Redis (modo cluster habilitado) ao qual você deseja adicionar fragmentos.

Tip

O Redis (modo cluster habilitado) exibe Clustered Redis (Redis clusterizado) na coluna Mode (Modo).

4. Escolha Add shard (Adicionar fragmento).
 - a. Para Number of shards to be added, escolha o número de fragmentos que deseja adicionar a este cluster.
 - b. Para Availability zone(s) (Zonas de disponibilidade), escolha No preference (Sem preferência) ou Specify availability zones (Especificar zonas de disponibilidade).
 - c. Se você optar por Specify availability zones (Especificar zonsa de disponibilidade), para cada nó em cada fragmento, selecione a zona de disponibilidade do nó na lista de zonas de disponibilidade.
 - d. Escolha Add (Adicionar).

Adição de fragmentos (AWS CLI)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Redis (modo cluster habilitado) adicionando fragmentos por meio da AWS CLI.

Use os parâmetros a seguir com `modify-replication-group-shard-configuration`.

Parâmetros

- `--apply-immediately`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `--replication-group-id`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `--node-group-count`: obrigatório. Especifica o número de fragmentos (grupos de nós) que devem existir quando a operação for concluída. Ao adicionar fragmentos, o valor de `--node-group-count` deve ser maior que o número atual de fragmentos.

Como opção, você pode especificar a zona de disponibilidade para cada nó no grupo de replicação usando `--resharding-configuration`.

- `--resharding-configuration`: opcional. Uma lista de zonas de disponibilidade preferenciais para cada nó em cada fragmento no grupo de replicação. Use este parâmetro somente se o valor de `--node-group-count` for maior que o número atual de fragmentos. Se esse parâmetro for omitido ao adicionar fragmentos, a Amazon ElastiCache selecionará as zonas de disponibilidade para os novos nós.

O exemplo a seguir reconfigura os espaços de chave em quatro fragmentos no cluster do Redis (modo cluster habilitado) `my-cluster`. O exemplo também especifica a zona de disponibilidade para cada nó em cada fragmento. A operação começa imediatamente.

Example Adicionar fragmentos

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --resharding-configuration \  
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" \  
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" \  
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" \  
  
```

```
"PreferredAvailabilityZones=us-east-2d,us-east-2c" \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
--replication-group-id my-cluster ^  
--node-group-count 4 ^  
--resharding-configuration ^  
  "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^  
  "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^  
  "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^  
  "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^  
--apply-immediately
```

Para obter mais informações, consulte [modify-replication-group-shard-configuration](#) na AWS CLI documentação.

Adicionar fragmentos (ElastiCache API)

Você pode usar a ElastiCache API para reconfigurar os fragmentos em seu cluster Redis (modo de cluster ativado) on-line usando a operação. `ModifyReplicationGroupShardConfiguration`

Use os parâmetros a seguir com `ModifyReplicationGroupShardConfiguration`.

Parâmetros

- `ApplyImmediately=true`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `ReplicationGroupId`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `NodeGroupCount`: obrigatório. Especifica o número de fragmentos (grupos de nós) que devem existir quando a operação for concluída. Ao adicionar fragmentos, o valor de `NodeGroupCount` deve ser maior que o número atual de fragmentos.

Como opção, você pode especificar a zona de disponibilidade para cada nó no grupo de replicação usando `ReshardingConfiguration`.

- `ReshardingConfiguration`: opcional. Uma lista de zonas de disponibilidade preferenciais para cada nó em cada fragmento no grupo de replicação. Use este parâmetro somente se o valor de `NodeGroupCount` for maior que o número atual de fragmentos. Se esse parâmetro for omitido

ao adicionar fragmentos, a Amazon ElastiCache selecionará as zonas de disponibilidade para os novos nós.

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Redis (modo de cluster ativado) adicionando fragmentos usando a API. ElastiCache

Example Adicionar fragmentos

O exemplo a seguir adiciona grupos de nós ao cluster Redis (modo cluster habilitado) `my-cluster` de modo que haja um total de quatro grupos de nós quando a operação for concluída. O exemplo também especifica a zona de disponibilidade para cada nó em cada fragmento. A operação começa imediatamente.

```
https://elasticache.us-east-2.amazonaws.com/  
  ?Action=ModifyReplicationGroupShardConfiguration  
  &ApplyImmediately=true  
  &NodeGroupCount=4  
  &ReplicationGroupId=my-cluster  
  
  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone  
east-2a  
  
  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  
  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone  
east-2b  
  
  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone  
east-2a  
  
  &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  
  &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone  
east-2d  
  
  &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone  
east-2d  
  
  &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone  
east-2c
```

```
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20171002T192317Z
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [ModifyReplicationGroupShardConfiguration](#) na Referência ElastiCache da API.

Remoção de fragmentos com refragmentação online

Você pode remover fragmentos do seu cluster Redis (modo de cluster ativado) usando a API AWS Management Console AWS CLI, ou ElastiCache .

Tópicos

- [Remoção de fragmentos \(console\)](#)
- [Remoção de fragmentos \(AWS CLI\)](#)
- [Removendo fragmentos \(ElastiCacheAPI\)](#)

Remoção de fragmentos (console)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Redis (modo cluster habilitado) removendo fragmentos por meio da 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 Redis (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 ter mais informações, consulte [Exclusão de um grupo de replicação](#). O procedimento a seguir descreve o processo para excluir um ou mais fragmentos.

Para remover fragmentos do seu cluster do Redis (modo cluster habilitado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).

3. Localize e escolha o nome, não a caixa à esquerda do nome do cluster, do cluster do Redis (modo cluster habilitado) do qual você deseja remover fragmentos.

 Tip

Os clusters do Redis (modo cluster habilitado) têm um valor igual ou superior a 1 na coluna Shards (Fragmentos).

4. Na lista de fragmentos, escolha a caixa à esquerda do nome de cada fragmento que você gostaria de excluir.
5. Escolha Delete shard (Excluir fragmento).

Remoção de fragmentos (AWS CLI)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Redis (modo cluster habilitado) removendo fragmentos por meio da AWS CLI.

 Important

Antes de remover grupos de nós (fragmentos) do seu grupo de replicação, certifique-se ElastiCache de que todos os seus dados caibam nos fragmentos restantes. Se os dados couberem, os fragmentos especificados (`--node-groups-to-remove`) serão excluídos do grupo de replicação como solicitado e seus espaços de chave serão mapeados para os fragmentos restantes. Se os dados não couberem nos grupos de nós restantes, o processo será encerrado e o grupo de replicação será deixado com a mesma configuração do grupo de nós anterior à solicitação.

Você pode usar o AWS CLI para remover um ou mais fragmentos do seu cluster Redis (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 ter 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 IDs de fragmentos (grupos de nós) a serem removidos do grupo de replicação.

O procedimento a seguir descreve o processo para excluir um ou mais fragmentos.

Example Remover fragmentos

O exemplo a seguir remove dois grupos de nós do cluster do Redis (modo cluster habilitado) `my-cluster` de modo que haja um total de dois grupos de nós quando a operação for concluída. Os espaços de chave dos fragmentos removidos são distribuídos uniformemente entre os fragmentos restantes.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 2 \  
  --node-groups-to-remove "0002" "0003" \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
  --replication-group-id my-cluster ^  
  --node-group-count 2 ^  
  --node-groups-to-remove "0002" "0003" ^  
  --apply-immediately
```

Removendo fragmentos (ElastiCacheAPI)

Você pode usar a ElastiCache API para reconfigurar os fragmentos em seu cluster Redis (modo de cluster ativado) on-line usando a operação `ModifyReplicationGroupShardConfiguration`

O processo a seguir descreve como reconfigurar os fragmentos em seu cluster Redis (modo de cluster ativado) removendo os fragmentos usando a API. ElastiCache

Important

Antes de remover grupos de nós (fragmentos) do seu grupo de replicação, certifique-se ElastiCache de que todos os seus dados caibam nos fragmentos restantes. Se os dados couberem, os fragmentos especificados (`NodeGroupsToRemove`) serão excluídos do grupo de replicação como solicitado e seus espaços de chave serão mapeados para os fragmentos restantes. Se os dados não couberem nos grupos de nós restantes, o processo será encerrado e o grupo de replicação será deixado com a mesma configuração do grupo de nós anterior à solicitação.

Você pode usar a ElastiCache API para remover um ou mais fragmentos do seu cluster Redis (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 ter 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 IDs de fragmentos (grupos de nós) a serem removidos do grupo de replicação.

O procedimento a seguir descreve o processo para excluir um ou mais fragmentos.

Example Remove fragmentos

O exemplo a seguir remove dois grupos de nós do cluster do Redis (modo cluster habilitado) `my-cluster` de modo que haja um total de dois grupos de nós quando a operação for concluída. Os espaços de chave dos fragmentos removidos são distribuídos uniformemente entre os fragmentos restantes.

```
https://elasticache.us-east-2.amazonaws.com/  
?Action=ModifyReplicationGroupShardConfiguration  
&ApplyImmediately=true  
&NodeGroupCount=2  
&ReplicationGroupId=my-cluster  
&NodeGroupsToRemove.member.1=0002  
&NodeGroupsToRemove.member.2=0003  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20171002T192317Z  
&X-Amz-Credential=<credential>
```

Rebalanceamento de fragmentos online

Você pode reequilibrar os fragmentos em seu cluster Redis (modo de cluster ativado) usando a API AWS Management Console AWS CLI, ou. ElastiCache

Tópicos

- [Rebalanceamento de fragmentos online \(Console\)](#)
- [Rebalanceamento de fragmentos online \(AWS CLI\)](#)
- [Rebalanceamento de fragmentos on-line \(API\) ElastiCache](#)

Rebalanceamento de fragmentos online (Console)

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Redis (modo cluster habilitado) rebalanceando fragmentos por meio da AWS Management Console.

Para rebalancear os espaços de chave entre os fragmentos no seu cluster do Redis (modo cluster habilitado)

1. Abra o ElastiCache console em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).

3. Escolha o nome do cluster (não a caixa à esquerda do nome do cluster) do Redis (modo cluster habilitado) que você gostaria de rebalancear.

 Tip

Os clusters do Redis (modo cluster habilitado) têm um valor igual ou superior a 1 na coluna Shards (Fragmentos).

4. Escolha Rebalance (Rebalancear).
5. Quando solicitado, escolha Rebalance (Rebalancear). Talvez você veja uma mensagem semelhante a esta: *Slots in the replication group are uniformly distributed (Os slots no grupo de replicação estão uniformemente distribuídos). Nada a fazer. (Serviço:AmazonElastiCache; Código de status: 400; Código de erro;; ID da solicitação: InvalidReplicationGroupState 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*. Se isso acontecer, escolha Cancel (Cancelar).

Rebalanceamento de fragmentos online (AWS CLI)

Use os parâmetros a seguir com `modify-replication-group-shard-configuration`.

Parâmetros

- `-apply-immediately`: obrigatório. Especifica que a operação de reconfiguração de fragmento começará imediatamente.
- `--replication-group-id`: obrigatório. Especifica em qual grupo de replicação (cluster) a operação de reconfiguração de fragmento será executada.
- `--node-group-count`: obrigatório. Para rebalancear os espaços de chave entre todos os fragmentos no cluster, esse valor deve ser igual ao número atual de fragmentos.

O processo a seguir descreve como reconfigurar os fragmentos no seu cluster do Redis (modo cluster habilitado) rebalanceando fragmentos por meio da AWS CLI.

Example - Rebalanceando os fragmentos em um cluster

O exemplo a seguir rebalanceia os slots no cluster do Redis (modo cluster habilitado) `my-cluster` de modo que os slots sejam distribuídos da forma mais uniforme possível. O valor de `--node-group-count` (4) é o número de fragmentos atualmente no cluster.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
  --replication-group-id my-cluster ^  
  --node-group-count 4 ^  
  --apply-immediately
```

Rebalanceamento de fragmentos on-line (API) ElastiCache

Você pode usar a ElastiCache API para reconfigurar os fragmentos em seu cluster Redis (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 Redis (modo de cluster ativado) reequilibrando os fragmentos usando a API `ElastiCache`.

Example - Rebalanceamento do cluster

O exemplo a seguir rebalanceia os slots no cluster do Redis (modo cluster habilitado) `my-cluster` de modo que os slots sejam distribuídos da forma mais uniforme possível. O valor de `NodeGroupCount` (4) é o número de fragmentos atualmente no cluster.

```
https://elasticache.us-east-2.amazonaws.com/  
?Action=ModifyReplicationGroupShardConfiguration  
&ApplyImmediately=true  
&NodeGroupCount=4  
&ReplicationGroupId=my-cluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20171002T192317Z  
&X-Amz-Credential=<credential>
```

Escalabilidade vertical online com modificação do tipo de nó

Usando a escalabilidade vertical online com o Amazon ElastiCache for Redis versão 3.2.10 ou posterior, você poderá escalar dinamicamente os clusters do Redis com tempo de inatividade mínimo. Isso permite que o cluster do Redis veicule solicitações mesmo ao ser dimensionado.

Note

Não há compatibilidade com escalabilidade entre um cluster de classificação de dados em níveis (p. ex., um cluster que use um tipo de nó r6gd) e um cluster sem classificação de dados em níveis (p. ex., um cluster que use um tipo de nó r6g). Para obter mais informações, consulte [Classificação de dados em níveis](#).

Você pode fazer o seguinte:

- Aumento de escala vertical: aumente a capacidade de leitura e gravação ajustando o tipo de nó do cluster do Redis para usar um tipo de nó maior.

O ElastiCache redimensiona dinamicamente o cluster ao permanecer online e atender a solicitações.

- Redução de escala vertical: reduza a capacidade de leitura e gravação ajustando o tipo de nó para usar um nó menor. Novamente, o ElastiCache redimensiona dinamicamente o cluster ao permanecer online e atender a 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:

- Verifique se você tem capacidade suficiente de ENI (interface de rede elástica). Se estiver reduzindo, garanta que o nó menor tenha memória suficiente para absorver o tráfego esperado.

Para ver as melhores práticas para o gerenciamento da memória, consulte [Gerenciamento da memória reservada](#).

- 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 on-line](#)
 - [Aumento de escala vertical de clusters de cache do Redis \(console\)](#)
 - [Aumento de escala vertical de clusters de cache do Redis \(AWS CLI\)](#)
 - [Aumento de escala vertical de clusters de cache do Redis \(API do ElastiCache\)](#)
- [Redução de escala vertical online](#)
 - [Redução de escala vertical de clusters de cache do Redis \(console\)](#)
 - [Redução de escala vertical de clusters de cache do Redis \(AWS CLI\)](#)
 - [Redução de escala vertical de clusters de cache do Redis \(API do ElastiCache\)](#)

Aumento de escala vertical on-line

Tópicos

- [Aumento de escala vertical de clusters de cache do Redis \(console\)](#)
- [Aumento de escala vertical de clusters de cache do Redis \(AWS CLI\)](#)

- [Aumento de escala vertical de clusters de cache do Redis \(API do ElastiCache\)](#)

Aumento de escala vertical de clusters de cache do Redis (console)

O procedimento a seguir descreve como expandir um cluster Redis usando o Console de Gerenciamento do ElastiCache. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Para aumentar a escala vertical de um cluster Redis (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).
3. Na lista de clusters, escolha o cluster.
4. Escolha Modify (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 Modify (Modificar).

Se você escolheu Apply immediately na etapa anterior, o status do cluster mudará para modifying. Quando o status mudar para available, a modificação estará completa, e você poderá começar a usar o novo cluster.

Aumento de escala vertical de clusters de cache do Redis (AWS CLI)

O procedimento a seguir descreve como expandir um cluster de cache Redis usando a AWS CLI. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Como aumentar a escala verticalmente um cluster de cache do Redis (AWS CLI)

1. Determine os tipos de nó para os quais você pode expandir executando o comando AWS CLI da `list-allowed-node-type-modifications` com o seguinte parâmetro.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-replication-group-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-replication-group-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ],  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small",  
    "cache.t2.medium",  
    "cache.t1.small"  
  ],  
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu grupo de replicação para expandir para o novo tipo de nó maior usando o comando `modify-replication-group` da AWS CLI com os seguintes parâmetros.

- `--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 semelhante a esta (formato JSON).

```
{
  "ReplicationGroup": {
    "Status": "modifying",
    "Description": "my-redis-cluster",
    "NodeGroups": [
      {
```

```

        "Status": "modifying",
        "Slots": "0-16383",
        "NodeGroupId": "0001",
        "NodeGroupMembers": [
            {
                "PreferredAvailabilityZone": "us-east-1f",
                "CacheNodeId": "0001",
                "CacheClusterId": "my-redis-cluster-0001-001"
            },
            {
                "PreferredAvailabilityZone": "us-east-1d",
                "CacheNodeId": "0001",
                "CacheClusterId": "my-redis-cluster-0001-002"
            }
        ]
    },
    "ConfigurationEndpoint": {
        "Port": 6379,
        "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "my-redis-cluster",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "07:30-08:30",
    "MemberClusters": [
        "my-redis-cluster-0001-001",
        "my-redis-cluster-0001-002"
    ],
    "CacheNodeType": "cache.m3.xlarge",
    "DataTiering": "disabled"
    "PendingModifiedValues": {}
}
}

```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou `--apply-immediately`, verifique o status do cluster de cache usando o comando `describe-cache-clusters` da AWS CLI com o seguinte parâmetro. Quando o status mudar para `available`, você poderá começar a usar o novo nó de cluster de cache maior.

Aumento de escala vertical de clusters de cache do Redis (API do ElastiCache)

O processo a seguir escala seu cluster de cache do tipo de nó atual para um novo tipo de nó maior usando a API do ElastiCache. Durante esse processo, o ElastiCache for Redis atualiza as entradas do DNS para que elas apontem para os novos nós. Por isso, você não precisa atualizar os endpoints no seu aplicativo. Para o Redis 5.0.5 e acima, você pode dimensionar clusters habilitados para failover automático enquanto o cluster permanece online e atende às solicitações recebidas. Nas versões 4.0.10 e anteriores, você perceberá uma breve interrupção de leituras e gravações em versões anteriores do nó primário enquanto a entrada de DNS é atualizada.

O tempo necessário para 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 aumentar a escala vertical de um cluster de cache do Redis (API do ElastiCache)

1. Determine quais tipos de nós você pode aumentar a escala vertical usando a ação `ListAllowedNodeTypeModifications` da API do ElastiCache com o seguinte parâmetro.
 - `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 Referência da API do Amazon ElastiCache.

2. Aumente a escala vertical do seu grupo de replicação atual para o novo tipo de nó usando a ação `ModifyReplicationGroup` da API do ElastiCache e com os seguintes parâmetros.
 - `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 ação `ListAllowedNodeTypeModifications` na etapa 1.

- `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 Referência da API do Amazon ElastiCache.

3. Se você usou `ApplyImmediately=true`, monitore o status do grupo de replicação usando a ação `DescribeReplicationGroups` da API do ElastiCache 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 Referência da API do Amazon ElastiCache.

Redução de escala vertical online

Tópicos

- [Redução de escala vertical de clusters de cache do Redis \(console\)](#)
- [Redução de escala vertical de clusters de cache do Redis \(AWS CLI\)](#)
- [Redução de escala vertical de clusters de cache do Redis \(API do ElastiCache\)](#)

Redução de escala vertical de clusters de cache do Redis (console)

O procedimento a seguir descreve como reduzir um cluster Redis usando o Console de Gerenciamento do ElastiCache. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Para reduzir a escala vertical de um cluster Redis (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Regis).
3. Na lista de clusters, escolha seu cluster preferido.
4. Escolha Modify (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 Modify (Modificar).

Se você escolheu Apply immediately na etapa anterior, o status do cluster mudará para modifying. Quando o status mudar para available, a modificação estará completa, e você poderá começar a usar o novo cluster.

Redução de escala vertical de clusters de cache do Redis (AWS CLI)

O procedimento a seguir descreve como reduzir um cluster de cache Redis usando a AWS CLI. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

Como reduzir a escala verticalmente de um cluster de cache do Redis AWS CLI

1. Determine os tipos de nó que você pode reduzir executando o comando `list-allowed-node-type-modifications` da AWS CLI com o seguinte parâmetro.

Para Linux, macOS ou Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-replication-group-id
```

Para Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-replication-group-id
```

A saída do comando acima é semelhante a esta (formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",
```

```
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ]

  "ScaleDownModifications": [
    "cache.t2.micro",
    "cache.t2.small ",
    "cache.t2.medium ",
    "cache.t1.small"
  ]
}
```

Para obter mais informações, consulte [list-allowed-node-type-modifications](#) na AWS CLI Referência.

2. Modifique seu grupo de replicação para reduzir para o novo tipo de nó menor usando o comando `modify-replication-group` da AWS CLI com os seguintes parâmetros.
 - `--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 semelhante a esta (formato JSON).

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "my-redis-cluster",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "Slots": "0-16383",  
        "NodeGroupId": "0001",  
        "NodeGroupMembers": [  
          {  
            "PreferredAvailabilityZone": "us-east-1f",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-001"  
          },  
          {  
            "PreferredAvailabilityZone": "us-east-1d",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-002"  
          }  
        ]  
      }  
    ],  
  },  
}
```

```
    "ConfigurationEndpoint": {
      "Port": 6379,
      "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "my-redis-cluster",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "07:30-08:30",
    "MemberClusters": [
      "my-redis-cluster-0001-001",
      "my-redis-cluster-0001-002"
    ],
    "CacheNodeType": "cache.t2.micro",
    "DataTiering": "disabled"
    "PendingModifiedValues": {}
  }
}
```

Para obter mais informações, consulte [modify-replication-group](#) na AWS CLI Referência.

3. Se você usou `--apply-immediately`, verifique o status do cluster de cache usando o comando `describe-cache-clusters` da AWS CLI com o seguinte parâmetro. Quando o status mudar para `available`, você poderá começar a usar o novo nó de cluster de cache menor.

Redução de escala vertical de clusters de cache do Redis (API do ElastiCache)

O processo a seguir escala seu grupo de replicação do tipo de nó atual para um novo tipo de nó menor usando a API do ElastiCache. Durante esse processo, o cluster do Redis continuará a atender solicitações com tempo de inatividade mínimo.

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.

Redução de escala vertical (API do ElastiCache)

1. Determine quais tipos de nós você pode reduzir a escala vertical usando a ação `ListAllowedNodeTypeModifications` da API do ElastiCache com o seguinte parâmetro.
 - `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 Referência da API do Amazon ElastiCache.

2. Reduza a escala vertical do seu grupo de replicação atual para o novo tipo de nó usando a ação `ModifyReplicationGroup` da API do ElastiCache e com os seguintes parâmetros.

- `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 ação `ListAllowedNodeTypeModifications` na etapa 1.
- `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 Referência da API do Amazon ElastiCache.

Conceitos básicos do JSON no ElastiCache para Redis

O ElastiCache para Redis é compatível com o JavaScript Object Notation (JSON), que é uma maneira simples e sem esquema de codificar conjuntos de dados complexos em clusters do Redis. É possível armazenar e acessar dados nativamente usando o formato JavaScript Object Notation (JSON) dentro de clusters do Redis e atualizar dados do JSON armazenados nesses clusters, sem precisar gerenciar código personalizado para serializá-los e desserializá-los.

Além de usar as operações da API do Redis para aplicações que operam em JSON, agora é possível recuperar e atualizar com eficiência partes específicas de um documento JSON sem precisar manipular todo o objeto. Isso pode melhorar a performance e reduzir os custos. Também é possível pesquisar o conteúdo do seu documento JSON usando a consulta JSONPath [Goessner-style](#) (estilo Goessner).

Depois que você criar um cluster com uma versão de mecanismo compatível, o tipo de dados do JSON e os comandos associados estarão disponíveis automaticamente. Isso é compatível com a API e o RDB com a versão 2 do módulo RedisJSON, para que você possa migrar facilmente aplicações Redis baseadas em JSON para o ElastiCache para Redis. Para obter mais informações sobre os comandos do Redis compatíveis, consulte [Comandos compatíveis com o JSON do Redis](#).

As métricas relacionadas ao JSON `JsonBasedCmds` e `JsonBasedCmdsLatency` são incorporadas ao CloudWatch para monitorar o uso desse tipo de dados. Para obter mais informações, consulte [Métricas do Redis](#).

Note

Para usar o JSON, é necessário estar executando o mecanismo do Redis versão 6.2.6 ou posterior.

Tópicos

- [Visão geral do tipo de dados do JSON do Redis](#)
- [Comandos compatíveis com o JSON do Redis](#)

Visão geral do tipo de dados do JSON do Redis

O ElastiCache para Redis é compatível com vários comandos do Redis para trabalhar com o tipo de dados do JSON. Veja a seguir uma visão geral do tipo de dados do JSON e uma lista detalhada dos comandos do Redis que são compatíveis.

Terminologia

Prazo	Descrição
Documento JSON	Refere-se ao valor de uma chave JSON do Redis.
Valor JSON	Refere-se a um subconjunto de um documento JSON, incluindo a raiz que representa o documento inteiro. Um valor poderia ser um contêiner ou uma entrada dentro de um contêiner.
Elemento JSON	Equivalente ao valor JSON.

Padrão compatível com JSON

O formato JSON é compatível com os padrão de intercâmbio de dados do JSON [RFC 7159](#) e [ECMA-404](#). O padrão UTF-8 [Unicode](#) é compatível com texto do JSON.

Elemento raiz

O elemento raiz pode ser de qualquer tipo de dados do JSON. Observe que na RFC 4627 anterior, somente objetos ou matrizes eram permitidos como valores raiz. Desde a atualização para o RFC 7159, a raiz de um documento JSON pode ser de qualquer tipo de dados do JSON.

Limite de tamanho de documentos

Os documentos JSON são armazenados internamente em um formato que é otimizado para rápido acesso e modificação. Esse formato normalmente resulta no consumo um pouco maior de memória do que a representação serializada equivalente do mesmo documento.

O consumo de memória por um único documento JSON é limitado a 64 MB, que é o tamanho da estrutura de dados na memória, não a string JSON. É possível verificar a quantidade de memória consumida por um documento JSON usando o comando `JSON.DEBUG MEMORY`.

ACLs JSON

- Semelhante às categorias existentes por tipo de dados (`@string`, `@hash` etc.), uma nova categoria `@json` foi adicionada para simplificar o gerenciamento do acesso a comandos e dados do JSON. Nenhum outro comando Redis existente é membro da categoria `@json`. Todos os comandos JSON impõem restrições e permissões de keypace ou de comando.
- Existem cinco categorias existentes do Redis ACL que são atualizadas para incluir os novos comandos JSON: `@read`, `@write`, `@fast`, `@slow` e `@admin`. A tabela a seguir indica o mapeamento de comandos JSON para as categorias apropriadas.

ACL

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		
JSON.NUMMULTBY		y	y		
JSON.OBJECTS	y		y		
JSON.OBJECTLEN	y		y		
JSON.RESP	y		y		

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.SET		y		y	
JSON.STRAPPEND		y	y		
JSON.STRLEN	y		y		
JSON.STRLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

Limite de profundidade de aninhamento

Quando um objeto ou matriz JSON tem um elemento que é outro objeto ou matriz JSON, diz-se que esse objeto interno ou matriz se “aninha” dentro do objeto ou matriz externa. O limite máximo de profundidade de aninhamento é 128. Qualquer tentativa de criar um documento que contenha uma profundidade de aninhamento maior que 128 será rejeitada com um erro.

Sintaxe de comando

A maioria dos comandos exige um nome de chave Redis como primeiro argumento. Alguns comandos também têm um argumento path (caminho). O argumento path (caminho) será padronizado para a raiz se for opcional e não fornecido.

Notação:

- Os argumentos obrigatórios são colocados entre colchetes angulares. Por exemplo: <key>
- Os argumentos opcionais são colocados dentro de colchetes. Por exemplo: [path]

- Os argumentos opcionais adicionais são indicados por reticências (“...”). Por exemplo: [json...]

Sintaxe de caminho

O JSON do Redis oferece suporte a dois tipos de sintaxes de path:

- Sintaxe aprimorada: veja abaixo a sintaxe JSONPath descrita por [Goessner](#), conforme mostrado na tabela a seguir. Reordenamos e modificamos as descrições na tabela para maior clareza.
- Sintaxe restrita - Tem recursos de consulta limitados.

Note

Os resultados de alguns comandos são sensíveis ao tipo de sintaxe de caminho usado.

Se um caminho de consulta começar com '\$', ele usará a sintaxe aprimorada. Caso contrário, a sintaxe restrita será usada.

Sintaxe aprimorada

Símbolo/Expressão	Descrição
\$	O elemento raiz.
. ou []	Operador filho.
..	Descida recursiva.
*	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.

Símbolo/Expressão	Descrição
?()	Aplica uma expressão de filtro (script) à matriz ou objeto atual.
()	Expressão de filtro.
@	Usado em expressões de filtro que consultam o nó atual que está sendo processado.
==	Igual a, usado em expressões de filtro.
!=	Não é igual a, usado em expressões de filtro.
>	Maior que, usado em expressões de filtro.
>=	Maior que ou igual a, usado em expressões de filtro.
<	Menor que, usado em expressões de filtro.
<=	Menor que ou igual a, usado em expressões de filtro.
&&	AND lógico, usado para combinar várias expressões de filtro.
	OR lógico, usado para combinar várias expressões de filtro.

Exemplos

Os exemplos a seguir têm como base o exemplo de dados XML de [Goessner](#), que modificamos acrescentando campos adicionais.

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",
```

```
    "price": 8.95,
    "in-stock": true,
    "sold": true
  },
  { "category": "fiction",
    "author": "Evelyn Waugh",
    "title": "Sword of Honour",
    "price": 12.99,
    "in-stock": false,
    "sold": true
  },
  { "category": "fiction",
    "author": "Herman Melville",
    "title": "Moby Dick",
    "isbn": "0-553-21311-3",
    "price": 8.99,
    "in-stock": true,
    "sold": false
  },
  { "category": "fiction",
    "author": "J. R. R. Tolkien",
    "title": "The Lord of the Rings",
    "isbn": "0-395-19395-8",
    "price": 22.99,
    "in-stock": false,
    "sold": false
  }
],
"bicycle": {
  "color": "red",
  "price": 19.95,
  "in-stock": true,
  "sold": false
}
}
```

Path	Descrição
<code>\$.store.book[*].author</code>	Os autores de todos os livros da loja.
<code>\$..author</code>	Todos os autores.

Path	Descrição
<code>\$.store.*</code>	Todos os membros da loja.
<code>\$["store"].*</code>	Todos os membros da loja.
<code>\$.store..price</code>	O preço de tudo na loja.
<code>\$..*</code>	Todos os membros recursivos da estrutura JSON.
<code>\$.book[*]</code>	Todos os livros.
<code>\$.book[0]</code>	O primeiro livro.
<code>\$.book[-1]</code>	O último livro.
<code>\$.book[0:2]</code>	Os dois primeiros livros.
<code>\$.book[0,1]</code>	Os dois primeiros livros.
<code>\$.book[0:4]</code>	Livros do índice 0 a 3 (o índice final não é inclusivo).
<code>\$.book[0:4:2]</code>	Livros no índice 0, 2.
<code>\$.book[?(@.isbn)]</code>	Todos os livros com um número ISBN.
<code>\$.book[?(@.price<10)]</code>	Todos os livros mais baratos que US\$ 10.
<code>'\$.book[?(@.price < 10)]'</code>	Todos os livros são mais baratos do que US \$ 10. (O caminho deverá estar entre aspas se contiver espaços em branco.)
<code>'\$.book[?(@["price"] < 10)]'</code>	Todos os livros mais baratos que US\$ 10.
<code>'\$.book[?(@.["price"] < 10)]'</code>	Todos os livros mais baratos que US\$ 10.
<code>\$.book[?(@.price>=10&&@.price<=100)]</code>	Todos os livros na faixa de preço entre US\$ 10 a US\$ 100, inclusive.

Path	Descrição
'\$.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.

Métricas relacionadas ao JSON

As seguintes métricas de informações JSON são fornecidas:

Informações	Descrição
json_total_memory_bytes	Memória total alocada para objetos JSON.
json_num_documents	Número total de documentos no Redis.

Para consultar as métricas principais, execute os seguintes exemplos de comando do Redis:

```
info json_core_metrics
```

Como o ElastiCache para Redis interage com o JSON

A seção a seguir descreve como o ElastiCache para Redis interage com o tipo de dados do JSON.

Precedência do operador

Ao avaliar expressões condicionais para filtragem, `&&`s têm precedência primeiro e, em seguida, `||`s são avaliadas, como é comum na maioria das linguagens. As operações dentro de parênteses são executadas primeiro.

Comportamento do limite máximo de aninhamento de caminho

O limite máximo de aninhamento de caminho no ElastiCache para Redis é 128. Por isso, um valor como `$.a.b.c.d...` só pode atingir 128 níveis.

Processamento de valores numéricos

O JSON não tem tipos de dados separados para números inteiros e de ponto flutuante. Todos eles são chamados de números.

Representações numéricas:

Quando um número JSON é recebido na entrada, ele é convertido em uma das duas representações binárias internas: um valor inteiro com sinal de 64 bits ou um ponto flutuante de fluxo de precisão dupla de IEEE de 64 bits. A string original e toda a sua formatação não serão retidas. Dessa forma, quando um número é gerado como parte de uma resposta JSON, ele é convertido da representação binária interna para uma string imprimível que usa regras genéricas de formatação. Essas regras podem resultar em uma string diferente da que foi recebida.

Comandos aritméticos NUMINCRBY e NUMMULTBY:

- Se ambos os números forem inteiros e o resultado estiver fora da faixa de `int64`, ele se tornará automaticamente um número IEEE de ponto flutuante de precisão dupla de 64 bits.
- Se pelo menos um dos números for um ponto flutuante, o resultado será um número IEEE ponto flutuante de precisão dupla de 64 bits.
- Se o resultado exceder a faixa de 64 bits IEEE dupla, o comando `OVERFLOW` retornará um erro.

Para obter uma lista detalhada dos comandos disponíveis, consulte [Comandos compatíveis com o JSON do Redis](#).

Filtragem direta de matriz

O ElastiCache para Redis filtra objetos de matriz diretamente.

Para dados como `[0, 1, 2, 3, 4, 5, 6]` e uma consulta de caminho como `$[?(@<4)]`, ou dados como `{"my_key": [0, 1, 2, 3, 4, 5, 6]}` e uma consulta de caminho como `$.my_key[?(@<4)]`, o ElastiCache para Redis retornaria `[1,2,3]` em ambas as circunstâncias.

Comportamento de indexação de matriz

O ElastiCache para Redis permite tanto os índices positivo quanto o negativo para matrizes. Para uma matriz de comprimento cinco, 0 consultaria o primeiro elemento, 1 o segundo, e assim por diante. Números negativos começam no fim da matriz, então -1 consultaria o quinto elemento, -2 o quarto elemento e assim por diante.

Para garantir um comportamento previsível para os clientes, o ElastiCache para Redis não arredonda os índices de matriz para baixo ou para cima, portanto, se você tiver uma matriz com um comprimento de 5, chamar o índice 5 ou superior, ou -6 ou menor, não produzirá um resultado.

Avaliação estrita da sintaxe

MemoryDB não permite caminhos JSON com sintaxe inválida, mesmo que um subconjunto do caminho contenha um caminho válido. Isso acontece para manter o comportamento correto para nossos clientes.

Comandos compatíveis com o JSON do Redis

O ElastiCache para Redis é compatível com os seguintes comandos JSON do Redis:

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 (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- json (obrigatório) - O valor JSON a ser anexado à matriz.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Erro SYNTAXERR se um dos argumentos de entradas json não for uma string JSON 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.
- Erro `SYNTAXERR` se um dos argumentos de entradas json não for uma string JSON válida.
- `NONEXISTENT` erro se o caminho não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.ARRINDEX

Procura a primeira ocorrência de um valor escalar JSON nas matrizes no caminho.

- Erros fora do intervalo são tratados arredondando o índice para o início e o fim da matriz.
- Se início > fim, retorna -1 (não encontrado).

Sintaxe

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

- **chave (obrigatório)** - Uma chave Redis do tipo de documento JSON.
- **path (obrigatório)**: um caminho JSON.
- **json-scalar (obrigatório)** - O valor escalar a ser pesquisado. JSON escalar se refere a valores que não são objetos ou matrizes. Ou seja, sstring, número, bbooleano, e nulo são valores escalares.
- **início (opcional)** - O índice inicial, inclusive. Assumirá o padrão de 0 se não for fornecido.
- **fim (opcional)** - O índice final, exclusivo. Assumirá o padrão de 0 se não for fornecido, o que significa que o último elemento está incluído. 0 ou -1 significa que o último elemento está incluído.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de números inteiros. Cada valor é o índice do elemento correspondente na matriz no caminho. O valor é -1, se não encontrado.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.

Se o caminho for uma sintaxe restrita:

- Inteiro, o índice do elemento correspondente ou -1 se não for encontrado.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.ARRINSERT

Insere um ou mais valores nos valores da matriz no caminho antes do índice.

Sintaxe

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

- **chave** (obrigatório) - Uma chave Redis do tipo de documento JSON.
- **path** (obrigatório): um caminho JSON.
- **índice** (obrigatório) -Um índice de matriz antes do qual os valores são inseridos.
- **json** (obrigatório) - O valor JSON a ser anexado à matriz.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor for uma matriz vazia, seu valor de retorno correspondente será nulo.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Erro `OUTOFBOUNDARIES` se o argumento índice estiver fora dos limites.

Se o caminho for uma sintaxe restrita:

- Inteiro, o novo comprimento da matriz.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.
- Erro `OUTOFBOUNDARIES` se o argumento índice estiver fora dos limites.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.ARRLEN

Obtém o comprimento dos valores da matriz no caminho.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o comprimento da matriz em cada caminho.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- 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 (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.
- índice (opcional) -A posição na matriz a partir da qual começar a exibir.
 - O padrão é -1 se não é fornecido, o que significa o último elemento.
 - O valor negativo significa posição do último elemento.
 - Os índices fora do limite são arredondados para seus respectivos limites de matriz.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings em massa que representam os valores exibidos em cada caminho.
- Se um valor for uma matriz vazia, seu valor de retorno correspondente será nulo.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.

Se o caminho for uma sintaxe restrita:

- String em massa que representa o valor JSON exibido.

- Nulo se a matriz estiver vazia.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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

JSON.ARRTRIM

Reduz uma matriz no caminho para que ela se torne uma subarray [start, end], ambos inclusivos.

- Se a matriz estiver vazia, não faça nada, retorne 0.
- Se início for < 0, trate-a como 0.

- Se fim for \geq tamanho (tamanho da matriz), trate-a como tamanho-1.
- Se início for \geq tamanho ou início for $>$ fim, esvazie a matriz e retorne 0.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- início (obrigatório) - O índice inicial, inclusive.
- fim (obrigatório) - O índice final, inclusive.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor for uma matriz vazia, seu valor de retorno correspondente será nulo.
- Se um valor não for uma matriz, seu valor de retorno correspondente será nulo.
- Erro `OUTOFBOUNDARIES` se um argumento índice estiver fora dos limites.

Se o caminho for uma sintaxe restrita:

- Inteiro, o novo comprimento da matriz.
- Nulo se a matriz estiver vazia.
- Erro `WRONGTYPE` se o valor no caminho não for uma matriz.
- Erro `OUTOFBOUNDARIES` se um argumento índice estiver fora dos limites.

Exemplos

Sintaxe do caminho aprimorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[[], ["a"], ["a", "b"], ["a", "b", "c"]]'  
OK  
127.0.0.1:6379> JSON.ARRTRIM k1 $[*] 0 1
```

```
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 2
127.0.0.1:6379> JSON.GET k1
"[[[],["a\""],["a\"","\b\""],["a\"","\b\"]]]"
```

Sintaxe do caminho restrita:

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

JSON.CLEAR

Limpa as matrizes ou um objeto no caminho.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

- Inteiro, o número de contêineres limpos.
- Limpar uma matriz ou objeto vazio conta como 1 contêiner limpo.
- Limpar um valor que não seja do contêiner retorna 0.

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]** - Informa o uso de memória em bytes de um valor JSON. O caminho assumirá o padrão da raiz se não for fornecido.
- **FIELDS <key> [path]** - Informa o número de campos no caminho do documento especificado. O caminho assumirá o padrão da raiz se não for fornecido. Cada valor JSON não contêiner conta como um campo. Objetos e matrizes contam recursivamente como um campo para cada um dos valores JSON que contêm. Cada valor de contêiner, exceto o contêiner raiz, conta como um campo adicional.
- **HELP** - Imprime mensagens de ajuda referentes ao comando.

Sintaxe

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

Depende do subcomando:

MEMORY

- Se o caminho for uma sintaxe aprimorada:
 - Retorna uma matriz de inteiros que representa o tamanho da memória (em bytes) do valor JSON em cada caminho.
 - Retornará uma matriz vazia se a chave Redis não existir.
- Se o caminho for uma sintaxe restrita:

- Retorna um número inteiro, tamanho da memória eo valor de JSON em bytes.
- Retornará nulo se a chave Redis não existir.

FIELDS

- Se o caminho for uma sintaxe aprimorada:
 - Retorna uma matriz de inteiros que representam o número de campos do valor JSON em cada caminho.
 - Retornará uma matriz vazia se a chave Redis não existir.
- Se o caminho for uma sintaxe restrita:
 - Retorna um número inteiro, número de campos do valor JSON.
 - Retornará nulo se a chave Redis não existir.

HELP - Retorna uma série de mensagens de ajuda.

Exemplos

Sintaxe do caminho aprimorada:

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

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

Sintaxe do caminho restrita:

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

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

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

JSON.DEL

Exclui valores JSON no caminho em uma chave de documento. Se o caminho é a raiz, é equivalente a excluir a chave do Redis.

Sintaxe

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

- **chave (obrigatório)** - Uma chave Redis do tipo de documento JSON.
- **path (opcional)**: um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

- Número de elementos excluídos.
- 0 se a chave Redis não existir.
- 0 se o caminho JSON for inválido ou não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.FORGET

Um alias de [JSON.DEL](#).

JSON.GET

Retorna o JSON serializado em um ou vários caminhos.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- INDENT/NEWLINE/SPACE (opcional) —Controla o formato da string do JSON retornada, isto é, "formatado para impressão". O valor padrão é uma string vazia. Ele pode ser anulado em qualquer combinação. Eles podem ser especificados em qualquer ordem.
- NOESCAPE - Opcional, permitido estar presente para compatibilidade com legado e não tem outro efeito.
- path (opcional): zero ou mais caminhos JSON, assumirá o padrão de raiz se nenhum for fornecido. Os argumentos do caminho devem ser colocados no final.

Return

Sintaxe do caminho aprimorada:

Se um caminho for fornecido:

- Retornará a string serializada de uma matriz de valores.
- Se nenhum valor for selecionado, o comando retornará uma matriz vazia.

Se vários caminhos forem fornecidos:

- Retornará um objeto JSON em formato de string, no qual cada caminho é uma chave.

- Se houver sintaxe mista de caminho aprimorado e restrito, o resultado estará de acordo com a sintaxe aprimorada.
- Se um caminho não existir, seu valor correspondente será uma matriz vazia.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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

JSON.MGET

Obtém JSONs serializados no caminho a partir de várias chaves de documento. Ele retorna nulo para uma chave ou caminho JSON não existente .

Sintaxe

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

- chave (obrigatório) - Uma ou mais chaves Redis do tipo documento.
- path (obrigatório): um caminho JSON.

Return

- Matriz de strings em massa. O tamanho da matriz é igual ao número de chaves no comando. Cada elemento da matriz será preenchido com (a) o JSON serializado conforme localizado pelo caminho ou (b) nulo se a chave não existir, o caminho não existir no documento ou o caminho for inválido (erro de sintaxe).
- Se alguma das chaves especificadas existir e não for uma chave JSON, o comando retornará o erro WRONGTYPE.

Exemplos

Sintaxe do caminho aprimorada:

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

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

Sintaxe do caminho restrita:

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

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

JSON.NUMINCRBY

Incrementa os valores numéricos no caminho por um determinado número.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- número (obrigatório) - Um número.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings em massa que representam os valores resultantes em cada caminho.
- Se um valor não for um número, seu valor de retorno correspondente será nulo.

- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de duplo IEEE de 64 bits.
- `NONEXISTENT` se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz de strings em massa que representam os valores resultantes em cada caminho.
- Se vários valores forem selecionados, o comando retornará o resultado do último valor atualizado.
- Erro `WRONGTYPE` se o valor no caminho não for um número.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de duplo IEEE de 64 bits.
- `NONEXISTENT` se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

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

127.0.0.1:6379> JSON.SET k2 $ '{"a":{ }, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
```

```

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

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

JSON.NUMMULTBY

Multiplica os valores numéricos no caminho por um determinado número.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (obrigatório): um caminho JSON.
- número (obrigatório) - Um número.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings em massa que representam o valor resultante em cada caminho.
- Se um valor não for um número, seu valor de retorno correspondente será nulo.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de um número IEEE de 64 bits do ponto flutuante de precisão dupla.
- `NONEXISTENT` se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz de strings em massa que representam os valores resultantes em cada caminho.
- Se vários valores forem selecionados, o comando retornará o resultado do último valor atualizado.
- Erro `WRONGTYPE` se o valor no caminho não for um número.
- Erro `WRONGTYPE` se o número não puder ser analisado.
- Erro `OVERFLOW` se o resultado estiver fora do intervalo de um número IEEE de 64 bits duplo.
- `NONEXISTENT` se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

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

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

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

Sintaxe do caminho restrita:

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

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

127.0.0.1:6379> JSON.SET k2 . '{"a":{ }, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 .a.* 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k2 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":2},\"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\":{\"a\":2},\"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\":{\"a\":2},\"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 (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o comprimento do objeto em cada caminho.
- Se um valor não for um objeto, seu valor de retorno correspondente será nulo.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, número de chaves no objeto.
- Se vários objetos forem selecionados, o comando retornará o comprimento do primeiro objeto.
- Erro `WRONGTYPE` se o valor no caminho não for um objeto.
- `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 (obrigatório)** - Uma chave Redis do tipo de documento JSON.
- **path (opcional)**: um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- **Matriz de matriz de strings em massa**. Cada elemento é uma matriz de chaves em um objeto correspondente.
- Se um valor não for um objeto, seu valor de retorno correspondente será vazio.

- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz de strings em massa. Cada elemento é um nome de chave no objeto.
- Se vários objetos forem selecionados, o comando retornará as chaves do primeiro objeto.
- Erro `WRONGTYPE` se o valor no caminho não for um objeto.
- `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 valor JSON em um determinado caminho fornecido no protocolo de Serialização Redis (RESP). Se o valor for contêiner, a resposta será uma matriz RESP ou matriz aninhada.

- JSON nulo é mapeado para o RESP Null Bulk String.
- Valores booleanos JSON são associados às respectivas strings simples RESP.
- Os números inteiros são mapeados para números inteiros RESP.
- Os números de ponto flutuante duplo IEEE de 64 bits são mapeados para RESP Bulk Strings.
- As strings JSON são mapeadas para RESP Bulk Strings.
- As matrizes JSON são representados como matrizes RESP, onde o primeiro elemento é a string simples [, seguida pelos elementos da matriz.
- Os objetos JSON são representados como matrizes RESP, onde o primeiro elemento é a string simples {, seguida por pares de valores-chave, cada um dos quais é uma string RESP em massa.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de matrizes. Cada elemento da matriz representa a forma RESP do valor em um caminho.
- Matriz vazia se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Matriz representa a forma RESP do valor em um caminho.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

```
127.0.0.1:6379> JSON.RESP k1 $.address
```

```
1) 1) {
  2) 1) "street"
     2) "21 2nd Street"
  3) 1) "city"
     2) "New York"
  4) 1) "state"
     2) "NY"
  5) 1) "zipcode"
     2) "10021-3100"
```

```
127.0.0.1:6379> JSON.RESP k1 $.address.*
```

```
1) "21 2nd Street"
2) "New York"
3) "NY"
4) "10021-3100"
```

```
127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers
```

```
1) 1) [
  2) 1) {
     2) 1) "type"
        2) "home"
     3) 1) "number"
        2) "555 555-1234"
  3) 1) {
     2) 1) "type"
        2) "office"
```

```

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

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

```

Sintaxe do caminho restrita:

```

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

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

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

```

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

JSON.SET

Define os valores JSON no caminho.

Se o caminho exigir um membro do objeto:

- Se o elemento pai não existir, o comando retornará um erro INEXISTENT.
- Se o elemento pai não existir, o comando retornará ERROR.
- Se o elemento pai existir e for um objeto:

- Se o membro não existir, um novo membro será anexado ao objeto pai se e somente se o objeto pai for o último filho no caminho. Caso contrário, o comando retornará um erro INEXISTENT.
- Se o membro existir, seu valor será substituído pelo valor JSON.

Se o caminho exigir um índice de matriz:

- Se o elemento pai não existir, o comando retornará um erro INEXISTENT.
- Se o elemento pai existir, mas não for uma matriz o comando retornará ERROR.
- Se o elemento pai existir, mas o índice estiver fora dos limites, o comando retornará um erro OUTFBOUNDARIES.
- Se o elemento pai existir e o índice for válido, o elemento será substituído pelo novo valor JSON.

Se o caminho solicitar um objeto ou matriz, o valor (objeto ou matriz) será substituído pelo novo valor JSON.

Sintaxe

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

[NX | XX] Onde é possível ter 0 ou 1 de identificadores [NX | XX].

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (obrigatório): um caminho JSON. Para uma nova chave do Redis, o caminho JSON deve ser a raiz “.”.
- NX (opcional) - Se o caminho for a raiz, defina o valor somente se a chave Redis 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 Redis 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) OUTFOUBOUNDARIES Array index is out of bounds
```

JSON.STRAPPEND

Anexa uma string às strings JSON no caminho.

Sintaxe

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

- **chave** (obrigatório) - Uma chave Redis do tipo de documento JSON.
- **path** (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.
- **json_string** (obrigatório) - A representação JSON de uma string. Observe que uma string JSON deve estar entre aspas. Por exemplo: "exemplo de string".

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o novo comprimento da matriz em cada caminho.
- Se um valor não for uma string, seu valor de retorno correspondente será nulo.
- Erro SYNTAXERR se o argumento da entrada json não for uma string JSON 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.
- Erro WRONGTYPE se o argumento da entrada json não for uma string JSON válida.
- NONEXISTENT erro se o caminho não existir.

Exemplos

Sintaxe do caminho aprimorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a", "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
```

```
OK
```

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a 'a'
```

```
1) (integer) 2
```

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* 'a'
```

```
1) (integer) 3
```

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* 'a'
```

```
1) (integer) 2
```

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

Sintaxe do caminho restrita:

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

JSON.STRLLEN

Obtém o comprimento dos valores da string JSON no caminho.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros que representam o comprimento do valor da string em cada caminho.
- Se um valor não for uma string, seu valor correspondente será nulo.
- Nulo se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- Inteiro, o comprimento da string.
- Se vários valores de string forem selecionados, o comando retornará o comprimento da primeira string.
- Erro `WRONGTYPE` se o valor no caminho não for uma string.
- `NONEXISTENT` erro se o caminho não existir.
- Nulo se a chave do documento não existir.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

JSON.TOGGLE

Altera valores booleanos entre verdadeiro e falso no caminho.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de inteiros (0 - falso, 1 - verdadeiro) que representam o valor booleano resultante em cada caminho.
- Se um valor não for um valor booleano, seu valor de retorno correspondente será nulo.
- NONEXISTENT se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- String (“verdadeiro”/“falso”) que representa o valor booleano resultante.
- NONEXISTENT se a chave do documento não existir.
- Erro WRONGTYPE se o valor no caminho não for um valor booleano.

Exemplos

Sintaxe do caminho aprimorada:

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

Sintaxe do caminho restrita:

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

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

```
"true"  
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable  
"false"
```

JSON.TYPE

Informa o tipo de valores em um determinado caminho.

Sintaxe

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

- chave (obrigatório) - Uma chave Redis do tipo de documento JSON.
- path (opcional): um caminho JSON. Assumirá o padrão da raiz se não for fornecido.

Return

Se o caminho for uma sintaxe aprimorada:

- Matriz de strings que representam o tipo de valor em cada caminho. O tipo é um destes {"null", "boolean", "string", "number", "integer", "object" e "array"}.
- Se um caminho não existir, seu valor de retorno correspondente será nulo.
- Matriz vazia se a chave do documento não existir.

Se o caminho for uma sintaxe restrita:

- String, tipo do valor
- Nulo se a chave do documento não existir.
- Nulo se o caminho JSON for inválido ou não existir.

Exemplos

Sintaxe do caminho aprimorada:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
```

```
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
2) number
3) string
4) boolean
5) null
6) object
7) array
```

Sintaxe do caminho restrita:

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

Marcação de recursos do ElastiCache

Para ajudar você a gerenciar seus clusters e outros recursos do ElastiCache, é possível atribuir seus próprios metadados a cada recurso na forma de tags. As tags permitem categorizar seus recursos da AWS de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando

you have many resources of the same type. It's possible to quickly identify a specific resource based on the tags you assigned to it. This topic describes tags and shows you how to create them.

Warning

As a recommended practice, we suggest that you don't include confidential data in your tags.

Conceitos Básicos de Tags

A tag is a label assigned to an AWS resource. Each tag consists of a key and an optional value, both defined by you. Tags let you categorize your AWS resources in different ways, for example, by purpose or by owner. For example, you can define a set of tags for your ElastiCache clusters that help you track the owner and the user group of each instance.

We recommend that you develop a set of tag keys that meet your needs for each type of resource. Using a consistent set of tag keys makes it easier for you to manage your resources. It's possible to search and filter resources according to the tags you add. For more information about how to implement an effective tagging strategy for resources, consult the [whitepaper da AWS, Tagging Best Practices](#) (Recommended tagging practices).

Tags don't have semantic meaning in ElastiCache and are interpreted strictly as a string of characters. In addition, tags aren't automatically assigned to your resources. It's possible to edit tag keys and values, and it's possible to remove tags from a resource at any time. It's possible to define the value of a tag as `null`. When you add a tag that has the same key as an existing tag on that resource, the new value replaces the old one. If you delete a resource, all tags on that resource are also deleted. In addition, if you add or delete tags in a replication group, all nodes in that replication group also have their tags added or removed.

You can work with tags using the AWS Management Console, the AWS CLI, and the ElastiCache API.

If you're using IAM, you can control which users in your AWS account have permission to create, edit, or delete tags. For more information, consult [Permissões em nível de recurso](#).

Recursos que podem ser marcados

Você pode usar tags na maioria dos recursos do ElastiCache 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, é possível aplicar tags aos recursos usando o [Tag Editor](#). 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 (Nome), mas ela não tem nenhum significado semântico ao serviço do ElastiCache.

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 API do Amazon ElastiCache, a CLI da AWS ou um SDK da AWS, você poderá usar o parâmetro Tags na ação relevante da API do ElastiCache para aplicar tags. Eles são:

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`
- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

A tabela a seguir descreve os recursos do ElastiCache que podem ser marcados e os recursos que podem ser marcados na criação usando a API do ElastiCache, a CLI da AWS ou um SDK da AWS.

Suporte à marcação para recursos do ElastiCache

Compatível com tags	Oferece suporte à marcação na criação
Sim	Sim

Note

Não é possível marcar Datastores globais.

Você pode aplicar permissões no nível do recurso com base em tags nas suas políticas do IAM para ações de API do ElastiCache que oferecem suporte à marcação durante a criação para implementar 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 podem ser usados para parâmetros padrão e grupos de segurança.

- 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 pode ter apenas um valor.
- Comprimento máximo da chave – 128 caracteres Unicode em UTF-8.
- Comprimento máximo do valor – 256 caracteres Unicode em UTF-8.
- Embora o 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 prefixo `aws :` é reservado para uso da AWS. 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 os recursos do ElastiCache em que você pode usar tags, consulte [Recursos que podem ser marcados](#).

Exemplo de marcação de recursos

- Criar um cache sem servidor usando tags

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis \  
  --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 redis \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Criação de um snapshot sem servidor com tags.

```
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.

Para esse caso, se você adicionar tags sob solicitação, mesmo que o grupo de replicação contenha tags, o snapshot receberá somente as tags da solicitação.

```
aws elasticache create-snapshot \  
--replication-group-id testing-tags \  
--snapshot-name bkp-testing-tags-rg \  
--tags Key="work",Value="foo"
```

Exemplos de políticas de controle de acesso baseadas em tags

1. Permissão de ação `AddTagsToResource` para um cluster somente se o cluster tiver a tag `Project=XYZ`.

```
{  
  "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. Negação da ação CopySnapshot 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

Quando você adiciona tags de alocação de custos aos seus recursos no Amazon ElastiCache, pode acompanhar os custos agrupando as despesas nas suas faturas por valores de tag de recursos.

Uma tag de alocação de custos do ElastiCache é um par de chaves/valores que você define e associa a um recurso do 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 tags de alocação de custos para organizar sua fatura da AWS para refletir sua própria estrutura de custos. Para isso, inscreva-se para obter sua conta da AWS com os valores de chave de 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, você

pode etiquetar 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)`. Você pode então ver o total dos seus custos do ElastiCache 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 custo do ElastiCache em nós do Redis. Ao adicionar, listar, modificar, copiar ou remover uma tag, a operação é aplicada somente ao nó especificado.

Características de tags de alocação de custos do ElastiCache

- As tags de alocação de custos são aplicadas aos recursos do ElastiCache que são especificados nas operações da CLI e da API como um ARN. O tipo de recurso será um "cluster".

ARN de exemplo: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- A chave de tags é o nome obrigatório da tag. O valor da string da chave pode ser de 1 a 128 caracteres Unicode e não pode ser prefixado com `aws:`. A string pode conter apenas o conjunto de letras Unicode, dígitos, espaços em branco, sublinhados (`_`), pontos finais (`.`), dois-pontos (`:`), barras invertidas (`\`), sinais de igualdade (`=`), sinais de adição (`+`), hífen (`-`) ou sinais de arroba (`@`).
- O valor da tag é o valor opcional da tag. O valor da string do valor pode ser de 1 a 256 caracteres Unicode e não pode ser prefixado com `aws:`. A string pode conter apenas o conjunto de letras Unicode, dígitos, espaços em branco, sublinhados (`_`), pontos finais (`.`), dois-pontos (`:`), barras invertidas (`\`), sinais de igualdade (`=`), sinais de adição (`+`), hífen (`-`) ou sinais de arroba (`@`).
- Um recurso do ElastiCache pode ter um máximo de 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`.

A AWS não aplica nenhum significado semântico às suas tags. As tags são interpretadas estritamente como cadeias de caracteres. A AWS não define automaticamente nenhuma tag em qualquer recurso do ElastiCache.

Gerenciamento das suas tags de alocação de custos usando a AWS CLI

Você pode usar a AWS CLI para adicionar, modificar ou remover tags de alocação de custos.

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Tags de alocação de custos são aplicadas em nós do ElastiCache for Redis. O nó que receberá tag é especificado usando um ARN (Nome de recurso da Amazon).

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Tópicos

- [Listagem de tags usando a AWS CLI](#)
- [Adição de tags usando a AWS CLI](#)
- [Modificação de tags usando a AWS CLI](#)
- [Remoção der tags usando a AWS CLI](#)

Listagem de tags usando a AWS CLI

Você pode usar a AWS CLI para listar tags em um recurso do ElastiCache existente usando a operação [list-tags-for-resource](#).

O código a seguir usa a AWS CLI para listar as tags no nó de Redis `my-cluster-001` no cluster `my-cluster` na região `us-west-2`.

Para Linux, macOS ou Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Para Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso.

```
{  
  "TagList": [  
    {  
      "Value": "10110",  
      "Key": "CostCenter"  
    },  
    {  
      "Value": "EC2",  
      "Key": "Service"  
    }  
  ]  
}
```

Se não houver tags no recurso, a saída será uma TagList vazia.

```
{  
  "TagList": []  
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [list-tags-for-resource](#).

Adição de tags usando a AWS CLI

Você pode usar a AWS CLI para adicionar tags a um recurso do ElastiCache existente usando a operação da CLI [add-tags-to-resource](#). Se a chave de tag não existir no recurso, a chave e o valor serão adicionados ao recurso. Se a chave já existir no recurso, o valor associado a essa chave será atualizado para o novo valor.

O código a seguir usa a AWS CLI para adicionar as chaves Service e Region com os valores elasticache e us-west-2, respectivamente, ao nó my-cluster-001 no cluster my-cluster na região us-west-2.

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 Service=elasticache,Region=us-west-2
```

```
--tags Key=Service,Value=elasticache \  
      Key=Region,Value=us-west-2
```

Para Windows:

```
aws elasticache add-tags-to-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^  
  --tags Key=Service,Value=elasticache ^  
        Key=Region,Value=us-west-2
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso após a operação, conforme mostrado a seguir.

```
{  
  "TagList": [  
    {  
      "Value": "elasticache",  
      "Key": "Service"  
    },  
    {  
      "Value": "us-west-2",  
      "Key": "Region"  
    }  
  ]  
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [add-tags-to-resource](#).

Você também pode usar a AWS CLI para adicionar tags a um cluster ao criar um novo cluster usando a operação [create-cache-cluster](#). Não é possível adicionar tags ao criar um cluster usando o console de gerenciamento do ElastiCache. Depois que o cluster é criado, você pode usar o console para adicionar tags ao cluster.

Modificação de tags usando a AWS CLI

Você pode usar a AWS CLI para modificar as tags em um nó em um cluster do ElastiCache for Redis.

Para modificar tags:

- Use [add-tags-to-resource](#) para adicionar uma nova tag e um valor ou para alterar o valor associado a uma tag existente.

- Use [remove-tags-from-resource](#) para remover tags especificadas do recurso.

A saída de qualquer operação será uma lista de tags e seus valores no cluster especificado.

Remoção der tags usando a AWS CLI

Você pode usar a AWS CLI para remover tags de um nó existente em um cluster do ElastiCache for Redis usando a operação [remove-tags-from-resource](#).

O código a seguir usa a AWS CLI para remover as etiquetas com as chaves Service e Region do nó `my-cluster-001` no cluster `my-cluster` na região `us-west-2`.

Para Linux, macOS ou Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \  
  --tag-keys PM Service
```

Para Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^  
  --tag-keys PM Service
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso após a operação, conforme mostrado a seguir.

```
{  
  "TagList": []  
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [remove-tags-from-resource](#).

Gerenciamento das suas tags de alocação de custos usando a API do ElastiCache

Você pode usar a API do ElastiCache para adicionar, modificar ou remover tags de alocação de custos.

Tags de alocação de custos são aplicadas em clusters do ElastiCache for Memcached. O cluster que receberá tag é especificado usando um ARN (Nome de recurso da Amazon).

Amostra de ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Tópicos

- [Listagem de tags usando a API do ElastiCache](#)
- [Adição de tags usando a API do ElastiCache](#)
- [Modificação de tags usando a API do ElastiCache](#)
- [Remoção de tags usando a API do ElastiCache](#)

Listagem de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para listar tags em um recurso existente usando a operação [ListTagsForResource](#).

O código a seguir usa a API do ElastiCache para listar as tags no recurso `my-cluster-001` na região `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ListTagsForResource  
  &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Version=2015-02-02  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Adição de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para adicionar tags a um cluster do ElastiCache existente usando a operação [AddTagsToResource](#). Se a chave de tag não existir no recurso, a chave e o valor serão adicionados ao recurso. Se a chave já existir no recurso, o valor associado a essa chave será atualizado para o novo valor.

O código a seguir usa a API do ElastiCache para adicionar as chaves `Service` e `Region` com os valores `elasticache` e `us-west-2`, respectivamente, ao recurso `my-cluster-001` na região `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=AddTagsToResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Tags.member.1.Key=Service  
&Tags.member.1.Value=elasticache  
&Tags.member.2.Key=Region  
&Tags.member.2.Value=us-west-2  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Para obter mais informações, consulte [AddTagsToResource](#) na Referência da API do Amazon ElastiCache.

Modificação de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para modificar as tags em um cluster do ElastiCache.

Para modificar o valor de uma tag:

- Use a operação [AddTagsToResource](#) para adicionar uma nova tag e um valor ou para alterar o valor de uma tag existente.
- Use [RemoveTagsFromResource](#) para remover tags do recurso.

A saída de qualquer operação será uma lista de tags e seus valores no recurso especificado.

Use [RemoveTagsFromResource](#) para remover tags do recurso.

Remoção de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para remover tags de um nó do ElastiCache for Redis existente usando a operação [RemoveTagsFromResource](#).

O código a seguir usa a API do ElastiCache para remover as tags com as chaves `Service` e `Region` do nó `my-cluster-001` no cluster `my-cluster` na região `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RemoveTagsFromResource
```

```
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&TagKeys.member.1=Service
&TagKeys.member.2=Region
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Usar a Lente do Well-Architected para o Amazon ElastiCache

Esta seção descreve o recurso Lente do Well-Architected para o Amazon ElastiCache, um conjunto de princípios de design e orientações para projetar workloads bem arquitetadas do ElastiCache.

- A Lente para o ElastiCache é um aditivo ao [AWS Well-Architected Framework](#).
- Cada pilar tem um conjunto de perguntas para ajudar a iniciar a discussão sobre uma arquitetura do ElastiCache.
 - Cada pergunta tem várias práticas principais, além de suas pontuações para fins de relatório.
 - Obrigatório: necessário antes de entrar em produção (a ausência representa alto risco).
 - Ideal: o melhor estado possível em que um cliente pode estar.
 - Bom: o que recomendamos para os clientes (a ausência representa risco médio).
- Terminologia do Well-Architected
 - [Componente](#): código, configuração e recursos da AWS que, juntos, atendem a um requisito. Os componentes interagem com outros componentes e geralmente são equivalentes a um serviço em arquiteturas de microsserviços.
 - [Workload](#): um conjunto de componentes que, juntos, agregam valor comercial. Exemplos de workloads são sites de marketing, sites de comércio eletrônico, back-ends de aplicações móveis, plataformas de análises etc.

Tópicos

- [Pilar Excelência operacional da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar Segurança da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar Confiabilidade da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar Eficiência de performance da Lente do Well-Architected para o Amazon ElastiCache](#)
- [Pilar Otimização de custos da Lente do Well-Architected para o Amazon ElastiCache](#)

Pilar Excelência operacional da Lente do Well-Architected para o Amazon ElastiCache

O foco do pilar Excelência operacional está na execução e no monitoramento de sistemas para agregar valor empresarial e melhorar continuamente processos e procedimentos. Os principais tópicos incluem automatizar mudanças, responder a eventos e definir padrões para gerenciar operações diárias.

Tópicos

- [EO 1: Como você entende e responde aos alertas e eventos acionados pelo seu cluster do ElastiCache?](#)
- [EO 2: Quando e como você dimensiona seus clusters existentes do ElastiCache?](#)
- [EO 3: Como você gerencia os recursos do seu cluster do ElastiCache e mantém o cluster atualizado?](#)
- [EO 4: Como você gerencia as conexões de clientes aos seus clusters do ElastiCache?](#)
- [EO 5: Como você implanta componentes do ElastiCache para uma workload?](#)
- [EO 6: Como você planeja e mitiga falhas?](#)
- [EO 7: Como solucionar problemas em eventos do mecanismo Redis?](#)

EO 1: Como você entende e responde aos alertas e eventos acionados pelo seu cluster do ElastiCache?

Introdução: ao operar clusters do ElastiCache, você tem a opção de receber notificações e alertas quando eventos específicos ocorrerem. Por padrão, o ElastiCache registra em log [eventos](#) relacionados aos seus recursos, como um failover, substituição de nó, operação de ajuste de escala, manutenção programada e muito mais. Cada evento inclui a data e hora, o nome e tipo da origem e uma descrição.

Benefício: ser capaz de entender e gerenciar os motivos subjacentes aos eventos que acionam os alertas gerados pelo seu cluster permite que você opere com mais eficiência e responda aos eventos de forma adequada.

- [Obrigatório] Analise os eventos gerados pelo ElastiCache no console do ElastiCache (depois de selecionar sua região) ou usando o comando [describe-events](#) da [Amazon Command Line Interface](#) (AWS CLI) e a [API do ElastiCache](#). Configure o ElastiCache para enviar notificações sobre eventos de cluster importantes usando o Amazon Simple Notification Service (Amazon

SNS). Usar o Amazon SNS com seus clusters permite que você realize ações programáticas em eventos do ElastiCache.

- Há duas grandes categorias de eventos: eventos atuais e programados. A lista de eventos atuais inclui: criação e exclusão de recursos, operações de ajuste de escala, failover, reinicialização de nó, criação de snapshot, modificação de parâmetros do cluster, renovação do certificado de CA, eventos de falha (falha no provisionamento do cluster [VPC ou ENI], falhas de ajuste de escala [ENI] e falhas de snapshot). A lista de eventos programados inclui: nó programado para substituição durante a janela de manutenção e substituição de nó reagendada.
- Embora talvez você não precise reagir imediatamente a alguns desses eventos, é fundamental examinar primeiro todos os eventos de falha:
 - ElastiCache:AddCacheNodeFailed
 - ElastiCache:CacheClusterProvisioningFailed
 - ElastiCache:CacheClusterScalingFailed
 - ElastiCache:CacheNodesRebooted
 - ElastiCache:SnapshotFailed (somente Redis)
- [Recursos]:
 - [Gerenciamento de notificações do Amazon SNS do ElastiCache](#)
 - [Notificações de eventos e o Amazon SNS](#)
- [Ideal] Para automatizar as respostas a eventos, utilize os recursos dos produtos e serviços da AWS, como o SNS e as funções do Lambda. Siga as práticas recomendadas ao fazer alterações pequenas, frequentes e reversíveis, como código para evoluir suas operações ao longo do tempo. Você deve usar as métricas do Amazon CloudWatch para monitorar seus clusters.

[Recursos]: [Monitorar endpoints de réplica de leitura do Amazon ElastiCache para Redis \(modo de cluster desabilitado\) usando o AWS Lambda, Amazon Route 53 e Amazon SNS](#) para conferir um caso de uso do Lambda com o SNS.

EO 2: Quando e como você dimensiona seus clusters existentes do ElastiCache?

Introdução: o dimensionamento correto do cluster do ElastiCache é um ato de equilíbrio que precisa ser avaliado sempre que houver alterações nos tipos de workload subjacentes. Seu objetivo é operar com o ambiente do tamanho certo para sua workload.

Benefício: a utilização excessiva de seus recursos pode resultar em latência elevada e diminuição geral da performance. Por outro lado, a subutilização pode resultar em provisionamento excessivo

de recursos com uma otimização de custos não ideal. Ao dimensionar corretamente seus ambientes, você pode encontrar um equilíbrio entre eficiência de performance e otimização de custos. Para remediar o excesso ou a subutilização de recursos, o ElastiCache pode ter a escala ajustada em duas dimensões. Você pode ajustar a escala verticalmente ao aumentar ou diminuir a capacidade dos nós. Também pode ajustar a escala horizontalmente ao adicionar e remover nós.

- [Obrigatório] O excesso de utilização da CPU e da rede nos nós primários deve ser resolvido descarregando e redirecionando as operações de leitura para os nós de réplica. Use nós de réplica para operações de leitura a fim de reduzir a utilização do nó primário. Isso pode ser configurado em sua biblioteca de cliente do Redis conectando-se ao endpoint de leitor do ElastiCache para desabilitar o modo de cluster ou usando o comando READONLY do Redis para habilitar o modo de cluster.

[Recursos]:

- [Localização de endpoints de conexão](#)
 - [Dimensionar um cluster corretamente](#)
 - [Comando READONLY do Redis](#)
- [Obrigatório] Monitore a utilização de recursos críticos do cluster, como CPU, memória e rede. A utilização desses recursos específicos do cluster precisa ser monitorada para informar sua decisão de ajustar a escala e do tipo de operação de ajuste. Com o modo de cluster do ElastiCache para Redis desabilitado, o nó primário e os nós de réplica podem escalar verticalmente. Os nós de réplica também podem escalar horizontalmente, de 0 a 5 nós. Com o modo de cluster habilitado, o mesmo se aplica a cada fragmento do cluster. Além disso, você pode aumentar ou reduzir o número de fragmentos.

[Recursos]:

- [Monitorar as práticas recomendadas com o Amazon ElastiCache para Redis usando o Amazon CloudWatch](#)
 - [Ajustar a escala de clusters do ElastiCache para Redis](#)
 - [Ajustar a escala de clusters do ElastiCache para Memcached](#)
- [Ideal] O monitoramento da performance ao longo do tempo pode ajudar você a detectar mudanças na workload que passariam despercebidas se monitoradas em momentos específicos. Para detectar tendências de longo prazo, use as métricas do CloudWatch a fim de verificar intervalos de tempo mais longos. O aprendizado da observação de longos períodos de métricas do CloudWatch deve informar sua previsão sobre a utilização de recursos do cluster. Os pontos de dados e métricas do CloudWatch ficam disponíveis por até 455 dias.

[Recursos]:

- [Monitorar o ElastiCache para Redis com métricas do CloudWatch](#)
 - [Monitorar o Memcached com métricas do CloudWatch](#)
 - [Monitorar as práticas recomendadas com o Amazon ElastiCache para Redis usando o Amazon CloudWatch](#)
- [Ideal] Se seus recursos do ElastiCache forem criados com o CloudFormation, uma prática recomendada é realizar alterações usando modelos do CloudFormation para preservar a consistência operacional e evitar alterações de configuração não gerenciadas e desvios de pilha.

[Recursos]:

- [Referência de tipos de recurso do ElastiCache para o CloudFormation](#)
- [Ideal] Automatize suas operações de ajuste de escala usando dados operacionais de cluster e defina limites no CloudWatch para configurar alarmes. Use o CloudWatch Events e o Simple Notification Service (SNS) para acionar funções do Lambda e executar uma API do ElastiCache a fim de ajustar a escala dos clusters automaticamente. Um exemplo seria adicionar um fragmento ao cluster quando a métrica `EngineCPUUtilization` atingir 80% por um longo período. Outra opção seria usar `DatabaseMemoryUsedPercentages` para um limite baseado em memória.

[Recursos]:

- [Usar alarmes do Amazon CloudWatch](#)
- [O que são os eventos do Amazon CloudWatch?](#)
- [Como usar o AWS Lambda com o Amazon Simple Notification Service](#)
- [Referência da API do ElastiCache](#)

EO 3: Como você gerencia os recursos do seu cluster do ElastiCache e mantém o cluster atualizado?

Introdução: ao operar em grande escala, é essencial que você seja capaz de localizar e identificar todos os seus recursos do ElastiCache. Ao lançar novos atributos da aplicação, você precisa criar simetria entre as versões do cluster em todos os tipos de ambiente do ElastiCache: desenvolvimento, teste e produção. Os atributos de recursos permitem que você separe ambientes para objetivos operacionais diferentes, como ao implantar novos atributos e ativar novos mecanismos de segurança.

Benefício: separar os ambientes de desenvolvimento, teste e produção é uma prática operacional recomendada. Também é uma prática recomendada que seus clusters e nós em todos os ambientes tenham os patches de software mais recentes aplicados usando processos bem compreendidos e documentados. Aproveitar os atributos nativos do ElastiCache permite que sua equipe de engenharia se concentre em cumprir os objetivos de negócios e não na manutenção do ElastiCache.

- [Ideal] Execute na versão mais recente do mecanismo disponível e aplique as atualizações de autoatendimento assim que estiverem disponíveis. O ElastiCache atualiza automaticamente sua infraestrutura subjacente durante a janela de manutenção especificada do cluster. No entanto, os nós em execução em seus clusters são atualizados por meio de atualizações de autoatendimento. Essas atualizações podem ser de dois tipos: patches de segurança ou pequenas atualizações de software. Compreenda a diferença entre os tipos de patches e quando eles são aplicados.

[Recursos]:

- [Atualizações de autoatendimento no Amazon ElastiCache](#)
- [Página de ajuda sobre manutenção gerenciada e atualizações de serviços do Amazon ElastiCache](#)
- [Ideal] Organize seus recursos do ElastiCache usando etiquetas. Use etiquetas em grupos de replicação e não em nós individuais. Você pode configurar etiquetas para serem exibidas ao consultar recursos e usar etiquetas para realizar pesquisas e aplicar filtros. Você deve usar grupos de recursos para criar e manter facilmente coleções de recursos que compartilham conjuntos comuns de etiquetas.

[Recursos]:

- [Práticas recomendadas de marcação](#)
- [Referência de tipos de recurso do ElastiCache para o CloudFormation](#)
- [Grupos de parâmetros](#)

EO 4: Como você gerencia as conexões de clientes aos seus clusters do ElastiCache?

Introdução: ao operar em grande escala, você precisa entender como seus clientes se conectam ao cluster do ElastiCache para gerenciar os aspectos operacionais da aplicação (como tempos de resposta).

Benefício: escolher o mecanismo de conexão mais adequado vai garantir que sua aplicação não se desconecte devido a erros de conectividade, como tempos limite.

- [Obrigatório] Separe as operações de leitura e gravação e conecte-se aos nós de réplica para executar as operações de leitura. No entanto, esteja ciente de que, ao separar as gravações das leituras, você perderá a capacidade de ler uma chave logo depois de gravá-la devido à natureza assíncrona da replicação do Redis. O comando WAIT pode ser usado para aumentar a segurança dos dados no mundo real e forçar as réplicas a reconhecer as gravações antes de responder aos clientes, com um custo geral de performance. O uso de nós de réplica para operações de leitura pode ser configurado em sua biblioteca de cliente do ElastiCache para Redis usando o endpoint de leitor do ElastiCache para o modo de cluster desabilitado. Para habilitar o modo de cluster, use o comando READONLY do ElastiCache para Redis. Para muitas das bibliotecas de cliente do ElastiCache para Redis, o READONLY do ElastiCache para Redis é implementado por padrão ou por meio de uma configuração.

[Recursos]:

- [Localização de endpoints de conexão](#)
- [READONLY](#)
- [Obrigatório] Use o agrupamento de conexões. Estabelecer uma conexão TCP tem um custo de tempo de CPU nos lados do cliente e do servidor, e o agrupamento permite que você reutilize a conexão TCP.

Para reduzir a sobrecarga da conexão, use o agrupamento de conexões. Com um grupo de conexões, sua aplicação pode reutilizar e liberar conexões “à vontade”, sem o custo de estabelecer a conexão. Você pode implementar o agrupamento de conexões por meio de sua biblioteca de cliente do ElastiCache para Redis (se compatível), com um framework disponível para o ambiente da sua aplicação, ou criá-lo do zero.

- [Ideal] Certifique-se de que o tempo limite do soquete do cliente esteja definido para pelo menos um segundo (em vez do padrão típico de “nenhum” em vários clientes).
 - Definir um valor de tempo limite muito baixo pode fazer com que o tempo limite seja atingido quando a carga do servidor estiver alta. Defini-lo muito alto pode fazer com que a aplicação demore muito para detectar problemas de conexão.
 - Controle o volume de novas conexões implementando o agrupamento de conexões em sua aplicação de cliente. Isso reduz a latência e a utilização da CPU necessárias para abrir e fechar conexões e realizar um handshake de TLS, caso o TLS esteja habilitado no cluster.

[Recursos]: [Configurar o Amazon ElastiCache para Redis para aumentar a disponibilidade](#)

- [Bom] Usar pipelines (quando seus casos de uso permitirem) pode aumentar significativamente a performance.

- Com pipelines, você reduz o tempo de ida e volta (RTT) entre os clientes da aplicação e o cluster, e novas solicitações podem ser processadas mesmo que o cliente ainda não tenha lido as respostas anteriores.
- Com pipelines, você pode enviar vários comandos para o servidor sem esperar por respostas ou confirmações. A desvantagem dos pipelines é que, quando você finalmente obtém todas as respostas em lote, pode ter ocorrido um erro que só será detectado no final.
- Implemente métodos para repetir as solicitações quando for retornado um erro que omite a solicitação incorreta.

[Recursos]: [Pipelines](#)

EO 5: Como você implanta componentes do ElastiCache para uma workload?

Introdução: os ambientes do ElastiCache podem ser implantados manualmente por meio do Console da AWS ou programaticamente por meio de APIs, CLI, kits de ferramentas etc. As práticas recomendadas de Excelência operacional sugerem que as implantações sejam automatizadas por meio de código sempre que possível. Além disso, os clusters do ElastiCache podem ser isolados por workload ou combinados para fins de otimização de custos.

Benefício: escolher o mecanismo de implantação mais adequado para seus ambientes do ElastiCache pode melhorar a excelência operacional ao longo do tempo. É recomendável realizar operações como código sempre que possível para minimizar a quantidade de erros humanos e aumentar a repetibilidade, a flexibilidade e o tempo de resposta aos eventos.

Ao entender os requisitos de isolamento da workload, você pode optar por ter ambientes dedicados do ElastiCache por workload ou combinar várias workloads em clusters únicos, ou combinações dessas estratégias. Compreender as vantagens e desvantagens pode ajudar a encontrar um equilíbrio entre Excelência operacional e Otimização de custos.

- [Obrigatório] Entenda as opções de implantação disponíveis para o ElastiCache e automatize esses procedimentos sempre que possível. Os possíveis caminhos de automação incluem CloudFormation, AWS CLI/SDK e APIs.

[Recursos]:

- [Referência de tipos de recurso do Amazon ElastiCache](#)
- [elasticache](#)
- [Referência da API do Amazon ElastiCache](#)

- [Obrigatório] Para todas as workloads, determine o nível de isolamento do cluster necessário.
 - [Ideal] Isolamento alto: mapeamento 1:1 entre workload e cluster. Permite o melhor controle sobre o acesso, o dimensionamento, a escalabilidade e o gerenciamento dos recursos do ElastiCache por workload.
 - [Melhor] Isolamento médio: M:1 isolado por finalidade, mas talvez compartilhado entre várias workloads (por exemplo, um cluster dedicado a armazenar workloads em cache e outro dedicado a mensagens).
 - [Bom] Isolamento baixo: M:1 multiuso, totalmente compartilhado. Recomendado para workloads em que o acesso compartilhado é aceitável.

EO 6: Como você planeja e mitiga falhas?

Introdução: a Excelência operacional inclui a antecipação de falhas por meio da realização de exercícios regulares “pré-mortem” a fim de identificar possíveis fontes de falha para que elas possam ser removidas ou mitigadas. O ElastiCache oferece uma API de failover que permite simular eventos de falha de nós, para fins de teste.

Benefício: ao testar cenários de falha com antecedência, você pode aprender como eles afetam sua workload. Isso permite testar com segurança os procedimentos de resposta e sua eficácia, além de familiarizar sua equipe com sua execução.

[Obrigatório] Realize regularmente testes de failover em contas de dev/teste. [TestFailover](#)

EO 7: Como solucionar problemas em eventos do mecanismo Redis?

Introdução: a Excelência operacional exige a capacidade de investigar as informações do serviço e do mecanismo para analisar a integridade e o status dos clusters. O Amazon ElastiCache para Redis pode emitir logs do mecanismo Redis para o Amazon CloudWatch e o Amazon Kinesis Data Firehose.

Benefício: habilitar os logs do mecanismo Redis nos clusters do Amazon ElastiCache para Redis fornece informações sobre eventos que afetam a integridade e a performance dos clusters. Os logs do mecanismo Redis fornecem dados diretamente do mecanismo Redis que não estão disponíveis por meio do mecanismo de eventos do ElastiCache. Por meio da observação cuidadosa dos eventos do ElastiCache (consulte a pergunta EO 1 anterior) e dos logs do mecanismo Redis, é possível determinar a ordem dos eventos ao solucionar problemas, tanto da perspectiva do serviço ElastiCache quanto da perspectiva do mecanismo Redis.

- [Obrigatório] Certifique-se de que a funcionalidade de registro em log do mecanismo Redis esteja habilitada. Ela está disponível no ElastiCache para Redis 6.2 e em versões posteriores. Isso pode ser feito durante a criação do cluster ou modificando o cluster depois da criação.
- Determine se o Amazon CloudWatch Logs ou o Amazon Kinesis Data Firehose são o destino apropriado para os logs do mecanismo Redis.
- Selecione um log de destino apropriado no CloudWatch ou no Kinesis Data Firehose para manter os logs. Se você tiver vários clusters, considere usar um log de destino diferente para cada cluster, pois isso ajudará a isolar os dados ao solucionar problemas.

[Recursos]:

- Entrega de logs: [Entrega de logs](#)
- Destinos de registro em log: [Amazon CloudWatch Logs](#)
- Introdução ao Amazon CloudWatch Logs: [O que é o Amazon CloudWatch Logs?](#)
- Introdução ao Amazon Kinesis Data Firehose: [O que é o Amazon Kinesis Data Firehose?](#)
- [Ideal] Se estiver usando o Amazon CloudWatch Logs, considere usar o Amazon CloudWatch Logs Insights para consultar o log do mecanismo Redis a fim de obter informações importantes.

Como exemplo, crie uma consulta ao grupo de logs do CloudWatch que contém os logs do mecanismo Redis que retornarão eventos com um LogLevel de "WARNING", como:

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Recursos]: [Analisar dados de logs com o CloudWatch Logs Insights](#)

Pilar Segurança da Lente do Well-Architected para o Amazon ElastiCache

O foco do pilar Segurança está na proteção de informações e sistemas. Os principais tópicos incluem confidencialidade e integridade dos dados, identificação e gerenciamento de quem pode fazer o quê com o gerenciamento baseado em privilégios, proteção de sistemas e estabelecimento de controles para detectar eventos de segurança.

Tópicos

- [SEG 1: Quais etapas você está tomando para controlar o acesso autorizado aos dados do ElastiCache?](#)

- [SEG 2: Suas aplicações exigem autorização adicional para o ElastiCache além dos controles baseados em rede?](#)
- [SEG 3: Existe o risco de que os comandos possam ser executados acidentalmente, causando perda ou falha de dados?](#)
- [SEG 4: Como garantir a criptografia de dados em repouso com o ElastiCache?](#)
- [SEG 5: Como criptografar dados em trânsito com o ElastiCache?](#)
- [SEG 6: Como restringir o acesso aos recursos do ambiente de gerenciamento?](#)
- [SEG 7: Como detectar e responder a eventos de segurança?](#)

SEG 1: Quais etapas você está tomando para controlar o acesso autorizado aos dados do ElastiCache?

Introdução: todos os clusters do ElastiCache são projetados para serem acessados de instâncias do Amazon Elastic Compute Cloud em uma VPC, de funções sem servidor (AWS Lambda) ou de contêineres (Amazon Elastic Container Service). O cenário mais comum é acessar um cluster do ElastiCache de uma instância do Amazon Elastic Compute Cloud na mesma Amazon VPC (Amazon Virtual Private Cloud). Antes de poder se conectar a um cluster usando uma instância do Amazon EC2, você deve autorizar a instância do Amazon EC2 a acessar o cluster. Para acessar um cluster do ElastiCache em execução em uma VPC, é necessário conceder entrada de rede ao cluster.

Benefício: a entrada de rede no cluster é controlada por meio de grupos de segurança da VPC. Um grupo de segurança atua como firewall virtual para as instâncias do Amazon EC2 a fim de controlar o tráfego de entrada e saída. As regras de entrada controlam o tráfego de entrada para a instância e as regras de saída controlam o tráfego de saída da instância. No caso do ElastiCache, ao iniciar um cluster é necessário associar um grupo de segurança. Isso garante que as regras de tráfego de entrada e saída estejam em vigor para todos os nós que compõem o cluster. Além disso, o ElastiCache está configurado para ser implantado exclusivamente em sub-redes privadas, de forma que elas só possam ser acessadas por meio de redes privadas da VPC.

- [Obrigatório] O grupo de segurança associado ao seu cluster controla a entrada na rede e o acesso ao cluster. Por padrão, um grupo de segurança não terá nenhuma regra de entrada definida e, portanto, nenhum caminho de entrada para o ElastiCache. Para habilitar isso, configure uma regra de entrada no grupo de segurança especificando o endereço/intervalo de IP de origem, o tráfego do tipo TCP e a porta do cluster do ElastiCache (porta padrão 6379 para o ElastiCache para Redis, por exemplo). Embora seja possível permitir um conjunto muito amplo de origens de entrada, como todos os recursos em uma VPC (0.0.0.0/0), é recomendável ser o mais

granular possível ao definir as regras de entrada, como autorizar somente o acesso de entrada a clientes Redis em execução em instâncias do Amazon EC2 associadas a um grupo de segurança específico.

[Recursos]:

- [Sub-redes e grupos de sub-redes](#)
- [Acessar um cluster ou grupo de replicação](#)
- [Controlar o tráfego para recursos usando grupos de segurança](#)
- [Grupos de segurança do Amazon Elastic Compute Cloud para instâncias do Linux](#)
- [Obrigatório] As políticas do AWS Identity and Access Management podem ser atribuídas a funções do AWS Lambda que permitam acessar os dados do ElastiCache. Para habilitar esse atributo, crie uma função de execução do IAM com a permissão `AWSLambdaVPCAccessExecutionRole` e atribua o perfil à função AWS Lambda.

[Recursos]: Configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC: [Tutorial: Configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC](#)

SEG 2: Suas aplicações exigem autorização adicional para o ElastiCache além dos controles baseados em rede?

Introdução: em cenários em que é necessário restringir ou controlar o acesso aos clusters do ElastiCache para Redis individualmente para cada cliente, é recomendável fazer a autenticação por meio do comando AUTH do ElastiCache para Redis. Os tokens de autenticação do ElastiCache para Redis, com gerenciamento opcional de usuários e grupos de usuários, possibilitam que o ElastiCache para Redis exija uma senha antes de permitir que os clientes executem comandos e chaves de acesso, aumentando assim a segurança do plano de dados.

Benefício: para ajudar a manter os dados em segurança, o ElastiCache para Redis fornece mecanismos de proteção contra o acesso não autorizado aos dados. Isso inclui impor que o AUTH do controle de acesso baseado em perfil (RBAC) ou um token de AUTH (senha) sejam usados pelos clientes para que se conectem ao ElastiCache antes de executar comandos autorizados.

- [Ideal] No ElastiCache para Redis 6.x ou superior, defina os controles de autenticação e autorização definindo grupos de usuários, usuários e strings de acesso. Atribua usuários a grupos de usuários, depois atribua grupos de usuários a clusters. Para utilizar o RBAC, ele deve ser

selecionado na criação do cluster e a criptografia em trânsito deve estar habilitada. Use um cliente Redis compatível com TLS para poder aproveitar o RBAC.

[Recursos]:

- [Aplicação do RBAC a um grupo de replicação para o ElastiCache para Redis](#)
- [Especificação de permissões usando uma string de acesso](#)
- [ACL](#)
- [Versões do ElastiCache para Redis com suporte](#)
- [Ideal] Em versões do ElastiCache para Redis anteriores à 6.x, além de definir token/senha fortes e manter uma política de senha rígida para AUTH do ElastiCache para Redis, a prática recomendada é alternar a senha/token. O ElastiCache pode gerenciar até dois (2) tokens de autenticação por vez. Você também pode modificar o cluster para exigir explicitamente o uso de tokens de autenticação.

[Recursos]: [Modificação do token AUTH em um cluster existente do ElastiCache para Redis](#)

SEG 3: Existe o risco de que os comandos possam ser executados acidentalmente, causando perda ou falha de dados?

Introdução: há vários comandos 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 comando FLUSHALL em um ambiente de desenvolvimento e, devido a um erro, pode tentar inadvertidamente chamar esse comando em um sistema de produção, resultando em perda acidental de dados.

Benefício: a partir do ElastiCache para Redis 5.0.3 no ElastiCache, é possível renomear certos comandos que podem atrapalhar sua workload. Renomear os comandos pode ajudar a evitar que sejam executados acidentalmente no cluster.

- [Obrigatório]

[Recursos]:

- [ElastiCache para Redis versão 5.0.3 \(obsoleto, use a versão 5.0.6\)](#)
- [Alterações de parâmetros do Redis 5.0.3](#)
- [Segurança do Redis](#)

SEG 4: Como garantir a criptografia de dados em repouso com o ElastiCache?

Introdução: embora o ElastiCache para Redis seja um datastore em memória, é possível criptografar qualquer dado que possa ser persistido (no armazenamento) como parte das operações padrão do cluster. Isso inclui backups programados e manuais gravados no Amazon S3, bem como dados salvos no armazenamento em disco como resultado de operações de sincronização e troca. Os tipos de instância nas famílias M6g e R6g também oferecem criptografia sempre ativa em memória.

Benefício: o ElastiCache para Redis fornece criptografia em repouso opcional para aumentar a segurança dos dados.

- [Obrigatório] A criptografia em repouso só pode ser habilitada em um cluster do ElastiCache (grupo de replicação) durante sua criação. Um cluster existente não pode ser modificado para começar a criptografar dados em repouso. Por padrão, o ElastiCache fornecerá e gerenciará as chaves usadas na criptografia em repouso.

[Recursos]:

- [Condições da criptografia em repouso](#)
- [Ativar criptografia em repouso](#)
- [Ideal] Utilize os tipos de instância do Amazon EC2 que criptografam dados enquanto eles estão na memória (como M6g ou R6g). Sempre que possível, considere gerenciar suas próprias chaves para criptografia em repouso. Para ambientes de segurança de dados mais rigorosos, o AWS Key Management Service (KMS) pode ser usado para autogerenciar as chaves mestras de cliente (CMK). Por meio da integração do ElastiCache com o AWS Key Management Service, você pode criar, possuir e gerenciar as chaves usadas para criptografia de dados em repouso em seu cluster do ElastiCache para Redis.

[Recursos]:

- [Uso de chaves gerenciadas pelo cliente do AWS Key Management Service](#)
- [AWS Key Management Service](#)
- [Conceitos do AWS KMS](#)

SEG 5: Como criptografar dados em trânsito com o ElastiCache?

Introdução: é um requisito comum evitar que os dados sejam comprometidos em trânsito. Isso representa dados dentro de componentes de um sistema distribuído, bem como entre clientes de aplicações e nós de cluster. O ElastiCache para Redis oferece suporte a esse requisito ao permitir a

criptografia de dados em trânsito entre clientes e cluster e entre os próprios nós do cluster. Os tipos de instância nas famílias M6g e R6g também oferecem criptografia sempre ativa em memória.

Benefício: a criptografia em trânsito do Amazon ElastiCache é um atributo opcional que permite aumentar a segurança de seus dados nos pontos mais vulneráveis: quando estão em trânsito de um local para outro.

- [Obrigatório] A criptografia em trânsito só pode ser habilitada em um cluster do ElastiCache para Redis (grupo de replicação) durante sua criação. Observe que, devido ao processamento adicional necessário para criptografar/descriptografar dados, a implementação da criptografia em trânsito vai afetar a performance. Para entender o impacto, é recomendável comparar a workload antes e depois de ativar a criptografia em trânsito.

[Recursos]:

- [Visão geral da criptografia em trânsito](#)

SEG 6: Como restringir o acesso aos recursos do ambiente de gerenciamento?

Introdução: as políticas do IAM e o ARN habilitam controles de acesso refinados no ElastiCache para Redis, permitindo um controle mais rígido para gerenciar a criação, modificação e exclusão de clusters do ElastiCache para Redis.

Benefício: o gerenciamento dos recursos do Amazon ElastiCache, como grupos de replicação, nós etc., pode ser restrito a contas da AWS que têm permissões específicas com base nas políticas do IAM, aumentando a segurança e a confiabilidade dos recursos.

- [Obrigatório] Gerencie o acesso aos recursos do Amazon ElastiCache atribuindo políticas do AWS Identity and Access Management específicas aos usuários da AWS, viabilizando um controle mais preciso sobre quais contas podem realizar quais ações nos clusters.

[Recursos]:

- [Visão geral do gerenciamento de permissões de acesso aos recursos do ElastiCache](#)
- [Uso de políticas baseadas em identidade \(políticas do IAM\) para o Amazon ElastiCache](#)

SEG 7: Como detectar e responder a eventos de segurança?

Introdução: quando implantado com o RBAC habilitado, o ElastiCache exporta métricas do CloudWatch para notificar os usuários sobre eventos de segurança. Essas métricas ajudam a

identificar tentativas fracassadas de autenticação, acesso a chaves ou execução de comandos para os quais os usuários do RBAC não têm autorização.

Além disso, os recursos de produtos e serviços da AWS ajudam a proteger a workload geral automatizando as implantações e registrando em log todas as ações e modificações para revisão/ auditoria posterior.

Benefício: ao monitorar eventos, sua organização consegue responder de acordo com seus requisitos, políticas e procedimentos. Automatizar o monitoramento e as respostas a esses eventos de segurança fortalece sua postura geral de segurança.

- [Obrigatório] Familiarize-se com as métricas publicadas do CloudWatch relacionadas às falhas de autenticação e autorização do RBAC.
 - `AuthenticationFailures` = tentativas fracassadas de autenticação no Redis
 - `KeyAuthorizationFailures` = tentativas fracassadas de usuários para acessar chaves sem permissão
 - `CommandAuthorizationFailures` = tentativas fracassadas de usuários para executar comandos sem permissão

[Recursos]:

- [Métricas para Redis](#)
- [Ideal] É recomendável configurar alertas e notificações sobre essas métricas e responder conforme necessário.

[Recursos]:

- [Usar alarmes do Amazon CloudWatch](#)
- [Ideal] Use o comando `ACL LOG` do Redis para obter mais detalhes.

[Recursos]:

- [ACL LOG](#)
- [Ideal] Familiarize-se com os recursos de produtos e serviços da AWS relacionados ao monitoramento, registro e análise de implantações e eventos do ElastiCache.

[Recursos]:

- [Logs das chamadas de API do Amazon ElastiCache com o AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-backup-check](#)
- [Monitorar o uso com métricas do CloudWatch](#)

Pilar Confiabilidade da Lente do Well-Architected para o Amazon ElastiCache

Tópicos

- [CONF 1: Como você está oferecendo suporte a implantações de arquitetura de alta disponibilidade \(HA\)?](#)
- [CONF 2: Como você está cumprindo seus objetivos de ponto de recuperação \(RPOs\) com o ElastiCache?](#)
- [CONF 3: Como você oferece suporte aos requisitos de recuperação de desastres \(DR\)?](#)
- [CONF 4: Como se planejar efetivamente para os failovers?](#)
- [CONF 5: Seus componentes do ElastiCache foram projetados para escalar?](#)

CONF 1: Como você está oferecendo suporte a implantações de arquitetura de alta disponibilidade (HA)?

Introdução: compreender a arquitetura de alta disponibilidade do Amazon ElastiCache permitirá que você opere em um estado resiliente durante eventos de disponibilidade.

Benefício: arquitetar seus clusters do ElastiCache para que sejam resilientes a falhas garante maior disponibilidade para suas implantações do ElastiCache.

- [Obrigatório] Determine o nível de confiabilidade necessário para seu cluster do ElastiCache. Workloads diferentes têm padrões de resiliência diferentes, desde workloads totalmente efêmeras até workloads essenciais à missão. Defina as necessidades de cada tipo de ambiente que você opera, como desenvolvimento, teste e produção.

Mecanismo de armazenamento em cache: Memcached versus ElastiCache para Redis

1. O Memcached não fornece nenhum mecanismo de replicação e é usado principalmente para workloads efêmeras.
 2. O ElastiCache para Redis oferece os atributos de HA discutidos abaixo.
- [Ideal] Para workloads que exigem HA, use o ElastiCache para Redis no modo de cluster com no mínimo duas réplicas por fragmento, mesmo para workloads com requisitos de throughput baixa que exijam apenas um fragmento.
 1. Com o modo de cluster habilitado, o multi-AZ é habilitado automaticamente.

O multi-AZ minimiza o tempo de inatividade realizando failovers automáticos do nó primário para as réplicas, em caso de manutenção planejada ou não planejada, além de mitigar falhas em AZ.

2. Para workloads fragmentadas, uma quantidade mínima de três fragmentos fornece uma recuperação mais rápida durante eventos de failover, pois o protocolo do Redis Cluster exige que a maioria dos nós primários esteja disponível para ter quórum.
3. Configure duas ou mais réplicas em toda a disponibilidade.

Ter duas réplicas proporciona maior escalabilidade de leitura e também disponibilidade de leitura em cenários em que uma réplica passa por manutenção.

4. Use tipos de nó baseados em Graviton2 (nós padrão na maioria das regiões).

O Amazon ElastiCache para Redis apresenta performance ainda mais otimizada nesses nós. Como resultado, você obtém melhor performance de replicação e sincronização, resultando em maior disponibilidade geral.

5. Monitore e ajuste o tamanho certo para lidar com picos de tráfego previstos: sob carga pesada, o mecanismo do ElastiCache para Redis pode deixar de responder, o que afeta a disponibilidade. BytesUsedForCache DatabaseMemoryUsagePercentage são bons indicadores do uso da memória, enquanto ReplicationLag é um indicador da integridade de replicação com base na taxa de gravação. Você pode usar essas métricas para acionar o ajuste de escala do cluster.
6. Garanta a resiliência do lado do cliente testando com a [API de failover antes de um evento de failover na produção](#).

[Recursos]:

- [Configurar o Amazon ElastiCache para Redis para aumentar a disponibilidade](#)
- [Alta disponibilidade com o uso de grupos de replicação](#)

CONF 2: Como você está cumprindo seus objetivos de ponto de recuperação (RPOs) com o ElastiCache?

Introdução: entenda o RPO da workload para embasar as decisões sobre estratégias de backup e recuperação do ElastiCache.

Benefício: ter uma estratégia de RPO em vigor pode melhorar a continuidade dos negócios nos cenários de recuperação de desastres. Projetar políticas de backup e restauração pode ajudar

o você a atingir seus objetivos de ponto de recuperação (RPO) para seus dados do ElastiCache. O ElastiCache para Redis oferece recursos de snapshot que são armazenados no Amazon S3, com uma política de retenção configurável. Esses instantâneos são gerados durante uma janela de backup definida e gerenciados automaticamente pelo serviço. Se sua workload exigir granularidade de backup adicional, você tem a opção de criar até 20 backups manuais por dia. Os backups criados manualmente não têm uma política de retenção de serviços e podem ser mantidos indefinidamente.

- [Obrigatório] Entenda e documente o RPO de suas implantações do ElastiCache.
 - Lembre-se de que o Memcached não oferece nenhum processo de backup.
 - Analise a capacidade dos atributos de backup e restauração do ElastiCache.
- [Ideal] Implemente um processo bem comunicado para fazer backup do cluster.
 - Inicie backups manuais conforme necessário.
 - Analise as políticas de retenção para backups automáticos.
 - Observe que os backups manuais serão mantidos indefinidamente.
 - Agende seus backups automáticos durante períodos de baixo uso.
 - Execute operações de backup em réplicas de leitura para garantir a minimização do impacto na performance do cluster.
- [Bom] Aproveite o atributo de backup agendado do ElastiCache para fazer backup regular de seus dados durante uma janela definida.
 - Teste periodicamente as restaurações de seus backups.
- [Recursos]:
 - [Redis](#)
 - [Backup e restauração do ElastiCache para Redis](#)
 - [Criação de backups manuais](#)
 - [Programação de backups automáticos](#)
 - [Backup e restauração de clusters do ElastiCache para Redis](#)

CONF 3: Como você oferece suporte aos requisitos de recuperação de desastres (DR)?

Introdução: a recuperação de desastres é um aspecto importante do planejamento de qualquer workload. O ElastiCache para Redis oferece várias opções para implementar a recuperação de desastres com base nos requisitos de resiliência da workload. Com o Amazon ElastiCache para

Redis Global Datastore, você pode gravar em seu cluster do ElastiCache para Redis em uma região e disponibilizar os dados para leitura em outros dois clusters de réplica entre regiões, viabilizando leituras de baixa latência e recuperação de desastres em várias regiões.

Benefício: compreender e se planejar para uma variedade de cenários de desastre pode garantir a continuidade dos negócios. As estratégias de DR devem equilibrar custo, impacto na performance e potencial de perda de dados.

- [Obrigatório] Desenvolva e documente estratégias de DR para todos os seus componentes do ElastiCache com base nos requisitos da workload. O ElastiCache é único porque alguns casos de uso são totalmente efêmeros e não exigem nenhuma estratégia de DR, enquanto outros estão na extremidade oposta do espectro e exigem uma estratégia de DR extremamente robusta. Todas as opções devem ser ponderadas em relação à otimização de custos: maior resiliência requer mais recursos de infraestrutura.

Entenda as opções de DR disponíveis em nível regional e multirregional.

- As implantações multi-AZ são recomendadas para evitar falhas de AZ. Realize a implantação com o modo de cluster habilitado em arquiteturas multi-AZ, com um mínimo de 3 AZs disponíveis.
- O Global Datastore é recomendado para se proteger contra falhas regionais.
- [Ideal] Habilite o Global Datastore para workloads que exigem resiliência por região.
 - Tenha um plano para realizar failover para a região secundária em caso de degradação da primária.
 - Teste o processo de failover multirregional antes de um failover na produção.
 - Monitore a métrica `ReplicationLag` para entender o impacto potencial da perda de dados durante eventos de failover.
- [Recursos]:
 - [Atenuar falhas](#)
 - [Replicação entre regiões da AWS usando datastores globais](#)
 - [Restauração de um backup com redimensionamento opcional do cluster](#)
 - [Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ](#)

CONF 4: Como se planejar efetivamente para os failovers?

Introdução: habilitar o multi-AZ com failovers automáticos é uma prática recomendada do ElastiCache. Em certos casos, o ElastiCache para Redis substitui os nós primários como parte das operações de serviço. Exemplos incluem eventos de manutenção planejada e o caso improvável de falha em um nó ou problema em zona de disponibilidade. Os failovers bem-sucedidos dependem da configuração do ElastiCache e da sua biblioteca de cliente.

Benefício: seguir as práticas recomendadas para failovers do ElastiCache em conjunto com sua biblioteca de cliente específica do ElastiCache para Redis ajuda a minimizar o tempo de inatividade potencial durante eventos de failover.

- [Obrigatório] Com o modo de cluster desabilitado, use tempos limite para que seus clientes detectem se precisam se desconectar do nó primário antigo e se reconectar ao novo nó primário, usando o endereço IP do endpoint primário atualizado. Com o modo de cluster habilitado, a biblioteca de cliente é responsável por detectar alterações na topologia subjacente do cluster. Isso é feito com mais frequência por meio de configurações na biblioteca de cliente do ElastiCache para Redis, que também permitem definir a frequência e o método de atualização. Cada biblioteca de cliente oferece configurações próprias e mais detalhes estão disponíveis na documentação correspondente.

[Recursos]:

- [Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ](#)
- Analise as práticas recomendadas da sua biblioteca de cliente do ElastiCache para Redis.
- [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 Redis, bem como as métricas do CloudWatch disponíveis para relatar o atraso na replicação entre o nó primário e os nós de réplica. Para casos de uso que exigem maior segurança de dados, use o comando WAIT do Redis para forçar as réplicas a reconhecerem as gravações antes de responder aos clientes conectados.

[Recursos]:

- [Métricas para Redis](#)
- [Monitorar as práticas recomendadas com o Amazon ElastiCache para Redis usando o Amazon CloudWatch](#)
- [Ideal] Valide regularmente a capacidade de resposta da aplicação durante o failover usando a API Test Failover do ElastiCache.

[Recursos]:

- [Testar o failover automático para uma réplica de leitura no Amazon ElastiCache para Redis](#)
- [Teste do failover automático](#)

CONF 5: Seus componentes do ElastiCache foram projetados para escalar?

Introdução: quando você entende os recursos de ajuste de escala e as topologias de implantação disponíveis, seus componentes do ElastiCache podem se ajustar ao longo do tempo para atender às mudanças nos requisitos da workload. O ElastiCache oferece ajuste de escala em quatro direções: aumento/redução na horizontal e na vertical.

Benefício: seguir as práticas recomendadas para implantações do ElastiCache oferece a maior flexibilidade para ajuste de escala, além de atender ao princípio do Well Architected de ajuste horizontal da escala para minimizar o impacto das falhas.

- **[Obrigatório]** Entenda a diferença entre topologias com modo de cluster habilitado e desabilitado. Em quase todos os casos, é recomendável realizar a implantação com o modo de cluster habilitado, pois isso aumenta a escalabilidade ao longo do tempo. Os componentes com modo de cluster desabilitado têm capacidade limitada de escalar horizontalmente com a adição de réplicas de leitura.
- **[Obrigatório]** Entenda quando e como escalar.
 - Para mais READIOPS: adicione réplicas.
 - Para mais WRITEOPS: adicione fragmentos (aumentar a escala horizontalmente).
 - Para mais E/S de rede: use instâncias otimizadas para rede (aumentar a escala verticalmente).
- **[Ideal]** Implante seus componentes do ElastiCache com o modo de cluster habilitado, com uma tendência para mais nós menores, em vez de menos nós maiores. Isso limita o raio de alcance de uma falha de nó.
- **[Ideal]** Inclua réplicas em seus clusters para melhorar a capacidade de resposta durante eventos de ajuste de escala.
- **[Bom]** Com o modo de cluster desabilitado, utilize as réplicas de leitura para aumentar a capacidade geral de leitura. O ElastiCache oferece suporte a até 5 réplicas de leitura com o modo de cluster desabilitado, além de ajuste vertical da escala.
- **[Recursos]:**
 - [Ajustar a escala de clusters do ElastiCache para Redis](#)

- [Aumento de escala vertical on-line](#)
- [Ajustar a escala de clusters do ElastiCache para Memcached](#)

Pilar Eficiência de performance da Lente do Well-Architected para o Amazon ElastiCache

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

- [EP 1: Como você monitora a performance de um cluster do Amazon ElastiCache?](#)
- [EP 2: Como você está distribuindo o trabalho entre os nós do cluster do ElastiCache?](#)
- [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?](#)
- [EP 6: Como você modela e interage com os dados no ElastiCache?](#)
- [EP 7: Como você registra comandos lentos em log no seu cluster do Amazon ElastiCache?](#)
- [EP 8: Como o ajuste de escala automático ajuda a aumentar a performance do cluster do ElastiCache?](#)

EP 1: Como você monitora a performance de um cluster do Amazon ElastiCache?

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 métricas do CloudWatch enquanto executa esses testes para compreender as métricas disponíveis e estabelecer uma referência de performance.
- [Ideal] Para workloads do ElastiCache para Redis, renomeie os comandos caros do ponto de vista computacional, como KEYS, para limitar a capacidade dos usuários de executar comandos de bloqueio em clusters de produção.
- As workloads do ElastiCache para Redis que executam o mecanismo 6.x podem aproveitar o controle de acesso baseado em perfil para restringir determinados comandos. O acesso aos comandos pode ser controlado por meio da criação de usuários e grupos de usuários com o Console da AWS ou a CLI, e associando os grupos de usuários a um cluster do ElastiCache para Redis. No Redis 6, quando o RBAC está habilitado, podemos usar “-@dangerous” para não permitir que determinado usuário execute comandos caros, como KEYS, MONITOR, SORT etc.
- Para o mecanismo versão 5.x, renomeie os comandos usando o parâmetro `rename-commands` no grupo de parâmetros do cluster do Amazon ElastiCache para Redis.
- [Melhor] Analise consultas lentas e procure técnicas de otimização.
 - Para workloads do ElastiCache para Redis, saiba mais sobre suas consultas analisando o log Lento. Por exemplo, você pode usar o comando `redis-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 do ElastiCache para Redis. 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 workloads do ElastiCache para Redis, `redis-benchmark` fornece uma interface simples para testar a performance 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]:
 - [Monitorar o uso com métricas do CloudWatch](#)
 - [Monitorar o uso com métricas do CloudWatch](#)
 - [Usar alarmes do Amazon CloudWatch](#)
 - [Parâmetros específicos do Redis](#)

- [SLOWLOG](#)
- [Referência do Redis](#)

EP 2: Como você está distribuindo o trabalho entre os nós do cluster do ElastiCache?

Introdução: a forma como sua aplicação se conecta aos nós do Amazon ElastiCache pode afetar a performance 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 endpoint adequado do ElastiCache.
 - O Amazon ElastiCache para Redis implementa diferentes endpoints com base no modo de cluster em uso. Com o modo de cluster habilitado, o ElastiCache fornecerá um endpoint de configuração. Com o modo de cluster desabilitado, o ElastiCache fornece um endpoint primário, normalmente usado para gravações, e um endpoint de leitor para balancear as leituras entre as 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 de vários nós do Memcached, o ElastiCache fornece um endpoint de configuração que habilita 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 o modo de cluster do ElastiCache para Redis habilitado para melhorar a escalabilidade.
 - Os clusters do ElastiCache para Redis (com modo de cluster habilitado) oferecem suporte a [operações de escalabilidade on-line](#) (aumento/redução da escala horizontal e verticalmente) para ajudar a distribuir os dados dinamicamente entre os 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 rebalancear o cluster movendo hashslots entre os fragmentos disponíveis no cluster do ElastiCache para Redis (com o modo de cluster habilitado). Isso ajuda a distribuir o trabalho de modo mais eficiente entre os fragmentos disponíveis.
- [Melhor] Implemente uma estratégia para identificar e corrigir chaves “hot” na workload.

- Considere o impacto das estruturas de dados multidimensionais do Redis, como listas, fluxos, conjuntos etc. Essas estruturas de dados são armazenadas em chaves únicas do Redis, 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 workloads do ElastiCache para Redis, você poderá detectar chaves “hot” usando `redis-cli --hotkeys` se uma política LFU de 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 próprio nó do Redis não fornecerá essa funcionalidade) e mantenha uma lista de nomes de chaves para leitura, além do nome da chave original.
- O ElastiCache para Redis versão 6 oferece suporte ao [armazenamento em cache do lado do cliente](#) auxiliado por servidor. Isso permite que as aplicações aguardem alterações em uma chave antes de fazerem chamadas de rede de volta para o ElastiCache.
- [Recursos]:
 - [Configurar o Amazon ElastiCache para Redis para aumentar a disponibilidade](#)
 - [Localização de endpoints de conexão](#)
 - [Práticas recomendadas para balanceamento de carga](#)
 - [Refragmentação e rebalanceamento de fragmentos online para o Redis \(modo cluster habilitado\)](#)
 - [Armazenamento em cache do lado do cliente no Redis](#)

EP 3: Para workloads de armazenamento em cache, como você monitora e relata a eficácia e a performance do cache?

Introdução: o armazenamento em cache é uma workload comum no ElastiCache e é importante que você entenda como gerenciar a eficácia e a performance do seu 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 Redis](#)
- [Obrigatório] Meça e colete a performance do cache da aplicação em conjunto com os valores de latência e utilização da CPU para entender se você precisa fazer ajustes no tempo de vida ou em outros componentes da aplicação. O ElastiCache fornece um conjunto de métricas do CloudWatch para latências agregadas para cada estrutura de dados. Essas métricas de latência são calculadas usando a estatística `commandstats` do comando `INFO` do ElastiCache para Redis e não incluem o tempo de rede e E/S. É apenas o tempo consumido pelo ElastiCache para Redis para processar as operações.

[Recursos]:

- [Métricas para Redis](#)
- [Monitorar as práticas recomendadas com o Amazon ElastiCache para Redis usando o 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](#)
- [Práticas recomendadas de armazenamento em cache](#)

- [Whitepaper Performance em grande escala com o Amazon ElastiCache](#)

EP 4: Como sua workload otimiza o uso de recursos e conexões de rede?

Introdução: vários clients de aplicações oferecem suporte ao ElastiCache para Redis e Memcached, 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 ao seu cluster do ElastiCache.
 - O agrupamento de conexões na aplicação reduz a sobrecarga criada no cluster devido à abertura e ao encerramento de conexões. Monitore o comportamento das conexões no Amazon CloudWatch usando `CurrConnections` e `NewConnections`.
 - 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]:
 - [Amazon ElastiCache para Redis: datastore global](#)
 - [Parâmetros específicos do Memcached](#)
 - [O Amazon ElastiCache para Redis 5.0.3 aprimora o tratamento de E/S para aumentar a performance](#)
 - [Métricas para Redis](#)
 - [Configurar o Amazon ElastiCache para Redis para aumentar a disponibilidade](#)

EP 5: Como você gerencia a exclusão e/ou remoção de chaves?

Introdução: as workloads têm requisitos e comportamentos esperados diferentes quando um nó de cluster está se aproximando dos limites de consumo de memória. O Amazon ElastiCache para Redis tem políticas diferentes para lidar com essas situações.

Benefício: o gerenciamento adequado da memória disponível e a compreensão das políticas de remoção vão ajudar a garantir o conhecimento do comportamento do cluster quando os limites de memória da instância forem excedidos.

- **[Obrigatório]** Instrumente o acesso aos dados para avaliar qual política aplicar. Identifique uma política de memória máxima apropriada para controlar se e como as remoções são realizadas no cluster.
 - A remoção ocorre quando a memória máxima do cluster é consumida e há uma política em vigor para permitir a remoção. O comportamento do cluster nessa situação depende da política de remoção especificada. Essa política pode ser gerenciada usando `maxmemory-policy` no grupo de parâmetros de cluster do ElastiCache para Redis.
 - A política padrão `volatile-lru` libera memória ao remover chaves com um prazo de validade definido (valor de TTL). As políticas menos usadas (LFU) e menos usadas recentemente (LRU) removem as chaves com base no uso.
 - Para workloads do Memcached, existe uma política LRU padrão que controla as remoções em cada nó. O número de remoções em seu cluster do Amazon ElastiCache pode ser monitorado usando a métrica Evictions no Amazon CloudWatch.
- **[Melhor]** Padronize o comportamento de exclusão para controlar o impacto na performance de seu cluster e evitar gargalos de performance inesperados.
 - Para workloads do ElastiCache para Redis, ao remover explicitamente as chaves do cluster, `UNLINK` funciona 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 workloads do ElastiCache para Redis 6.x, o comportamento do comando `DEL` pode ser modificado no grupo de parâmetros usando o parâmetro `lazyfree-lazy-user-del`.
- **[Recursos]:**
 - [Configuração de parâmetros do mecanismo usando grupos de parâmetros](#)
 - [UNLINK](#)
 - [Gerenciamento financeiro na nuvem com a AWS](#)

EP 6: Como você modela e interage com os dados no ElastiCache?

Introdução: o ElastiCache depende muito da aplicação das estruturas de dados e do modelo de dados usados, mas também precisa considerar o datastore subjacente (se houver). Entenda as estruturas de dados do ElastiCache para Redis disponíveis e garanta que você esteja usando as estruturas de dados mais adequadas às suas necessidades.

Benefício: a modelagem de dados no ElastiCache tem várias camadas, incluindo caso de uso de aplicação, tipos de dados e relacionamentos entre elementos de dados. Além disso, cada tipo de dado e comando do ElastiCache para Redis tem suas próprias assinaturas de performance bem documentadas.

- [Ideal] Uma das práticas recomendadas é reduzir a substituição não intencional de dados. Use uma convenção de nomenclatura que minimize a sobreposição de nomes de chave. A nomenclatura convencional de suas estruturas de dados usa um método hierárquico, como `APPNAME:CONTEXT:ID` e `ORDER-APP:CUSTOMER:123`.

[Recursos]:

- [Nomenclatura de chaves](#)
- [Ideal] Os comandos do ElastiCache para Redis têm uma complexidade temporal 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 única do ElastiCache para Redis, um grande volume de operações de alta complexidade temporal resultará em menor performance e em possíveis tempos de espera de operação.

[Recursos]:

- [Comandos](#)
- [Ideal] Use APIs para obter visibilidade da GUI no modelo de dados em seu cluster.

[Recursos]:

- [Comandante do Redis](#)
- [Redis Browser](#)
- [Redsmin](#)

EP 7: Como você registra comandos lentos em log no seu cluster do Amazon ElastiCache?

Introdução: benefícios do ajuste de performance por meio da captura, agregação e notificação de comandos de longa execução. Ao entender quanto tempo leva para que os comandos sejam executados, você pode determinar quais comandos resultam em performance ruim, bem como comandos que impedem a performance ideal do mecanismo. O Amazon ElastiCache para Redis também tem a capacidade de encaminhar essas informações para o 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 para Redis com mecanismo versão 6.0 ou mais recente, grupo de parâmetros configurado corretamente e registro em log SLOWLOG habilitado no cluster.
 - Os parâmetros necessários só estão disponíveis quando a compatibilidade com versões do mecanismo está definida como Redis versão 6.0 ou superior.
 - O registro em log SLOWLOG ocorre quando o tempo de execução de um comando pelo servidor é maior que um valor especificado. O comportamento do cluster depende dos parâmetros do grupo de parâmetros associado, que são `slowlog-log-slower-than` e `slowlog-max-len`.
 - As alterações terão efeito imediatamente.
- [Ideal] Aproveite os recursos do CloudWatch ou do Kinesis Data Firehose.
 - Use os recursos de filtragem e alarme do CloudWatch, CloudWatch Logs Insights e Amazon Simple Notification Services para implementar monitoramento de performance e notificação de eventos.
 - Use os recursos de streaming do Kinesis Data Firehose para arquivar logs SLOWLOG no armazenamento permanente ou para acionar o ajuste automático de parâmetros do cluster.
 - Determine qual formato atende melhor às suas necessidades, JSON ou texto simples.
 - Forneça permissões do IAM para publicar no CloudWatch ou no Kinesis Data Firehose.
- [Melhor] Configure `slowlog-log-slower-than` com um valor diferente do padrão.
 - Esse parâmetro determina por quanto tempo um comando pode ser executado no mecanismo Redis antes de ser registrado em log como um comando lento. O valor padrão é 10.000 microssegundos (10 milissegundos). O valor padrão pode ser muito alto para algumas workloads.

- Determine um valor que seja mais adequado para sua workload com base nas necessidades da aplicação e nos resultados dos testes, mas lembre-se de que um valor muito baixo pode gerar dados em excesso.
- [Melhor] Mantenha o valor padrão de `slowlog-max-len`.
 - Esse parâmetro determina o limite superior de quantos comandos lentos são capturados na memória do Redis por vez. O valor 0 desabilita a captura. Quanto maior o valor, mais entradas serão armazenadas na memória, reduzindo a chance de informações importantes serem removidas antes que possam ser revisadas. O valor padrão é 128.
 - O valor padrão é adequado para a maioria das workloads. Se houver necessidade de analisar dados em uma janela de tempo expandida na `redis-cli` por meio do comando `SLOWLOG`, considere aumentar esse valor. Isso permite que mais comandos permaneçam na memória do Redis.

Se você estiver emitindo os dados de `SLOWLOG` para o CloudWatch Logs ou para o Kinesis Data Firehose, os dados persistirão e poderão ser analisados fora do sistema do ElastiCache, reduzindo a necessidade de armazenar um grande número de comandos lentos na memória do Redis.

- [Recursos]:
 - [Como faço para ativar o log Lento do Redis em um cluster de cache do ElastiCache para Redis?](#)
 - [Entrega de logs](#)
 - [Parâmetros específicos do Redis](#)
 - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
 - [Amazon Kinesis Data Firehose](#)

EP 8: Como o ajuste de escala automático ajuda a aumentar a performance do cluster do ElastiCache?

Introdução: ao implementar o atributo de ajuste de escala automático do Redis, seus componentes do ElastiCache podem se ajustar ao longo do 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: a compreensão e o planejamento dos picos na workload podem garantir melhor performance do cache e a continuidade dos negócios. O ajuste de escala automático do ElastiCache

para Redis monitora continuamente a utilização da CPU/memória para garantir que seu cluster esteja operando nos níveis de performance desejados.

- [Obrigatório] Ao iniciar um cluster do ElastiCache para Redis:
 1. Verifique se o modo de cluster está habilitado.
 2. Verifique se a instância pertence a uma família de determinado tipo e tamanho compatíveis com o ajuste de escala automático.
 3. Verifique se o cluster não está sendo executado em datastores globais, Outposts ou zonas locais.

[Recursos]:

- [Escalabilidade de clusters no Redis \(modo cluster habilitado\)](#)
- [Usar o ajuste de escala automático com fragmentos](#)
- [Usar o ajuste de escala automático com réplicas](#)
- [Ideal] Identifique se sua workload exige muita leitura ou gravação para definir a política de ajuste de escala. Para obter a melhor performance, use apenas uma métrica de rastreamento. É recomendável evitar várias políticas para cada dimensão, pois as políticas de ajuste de escala automático aumentam a escala horizontalmente quando a meta é atingida, mas só aumentam a escala verticalmente quando todas as políticas de rastreamento de metas estiverem prontas para aumentar a escala verticalmente.

[Recursos]:

- [Políticas de ajuste de escala automático](#)
- [Definir uma política de ajuste de escala](#)
- [Ideal] O monitoramento da performance ao longo do tempo pode ajudar você a detectar mudanças na workload que passariam despercebidas se monitoradas em momentos específicos. Você pode analisar as métricas correspondentes do CloudWatch para utilização do cluster ao longo de 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]:

- [Monitorar o uso com métricas do CloudWatch](#)

- [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](#)
- [Registrar uma meta estável](#)

Pilar Otimização de custos da Lente do Well-Architected para o Amazon ElastiCache

O foco do pilar Otimização de custos é evitar custos desnecessários. Os principais tópicos incluem entender e controlar onde o dinheiro está sendo gasto, selecionar o tipo de nó mais adequado (usar instâncias que compatíveis com a hierarquização de dados com base nas necessidades da workload), a quantidade certa de tipos de recurso (quantidade de réplicas de leitura), a análise de gastos ao longo do tempo e a escalabilidade para atender às necessidades da empresa sem gastos excessivos.

Tópicos

- [CUSTOS 1: Como você identifica e monitora os custos associados aos seus recursos do ElastiCache? Como você desenvolve mecanismos para permitir que os usuários criem, gerenciem e descartem os recursos criados?](#)
- [CUSTOS 2: Como você usa ferramentas de monitoramento contínuo para ajudar a otimizar os custos associados aos seus recursos do ElastiCache?](#)
- [CUSTOS 3: Você deve usar um tipo de instância que ofereça suporte à hierarquização de dados? Quais são as vantagens de uma hierarquização de dados? Quando não usar instâncias de hierarquização de dados?](#)

CUSTOS 1: Como você identifica e monitora os custos associados aos seus recursos do ElastiCache? Como você desenvolve mecanismos para permitir que os usuários criem, gerenciem e descartem os recursos criados?

Introdução: entender as métricas de custo requer a participação e a colaboração de várias equipes: engenharia de software, gerenciamento de dados, proprietários de produtos, finanças e liderança. A identificação dos principais fatores de custo exige que todas as partes envolvidas compreendam

as alavancas de controle do uso do serviço e as vantagens e desvantagens do gerenciamento de custos, e essa costuma ser a principal diferença entre ações bem-sucedidas e não tão bem-sucedidas de otimização de custos. Garantir que você tenha processos e ferramentas em vigor para monitorar os recursos criados desde o desenvolvimento até a produção e a retirada ajuda a gerenciar os custos associados ao ElastiCache.

Benefício: o acompanhamento contínuo de todos os custos associados à workload exige uma compreensão profunda da arquitetura que inclui o ElastiCache como um de seus componentes. Além disso, você deve ter um plano de gerenciamento de custos para coletar e comparar o uso com seu orçamento.

- [Obrigatório] Institua um Centro de Excelência da Nuvem (CCoE) com um de seus estatutos fundadores para definir, monitorar e agir com base nas métricas relacionadas ao uso do ElastiCache em suas organizações. Se existir um CCoE funcional, garanta que ele saiba como ler e acompanhar os custos associados ao ElastiCache. Ao criar recursos, use políticas e perfis do IAM para validar que somente equipes e grupos específicos possam instanciar recursos. Isso vai garantir que os custos estejam associados aos resultados comerciais e que uma linha clara de responsabilidade seja estabelecida, do ponto de vista dos custos.
 1. O CCoE deve identificar, definir e publicar métricas de custo que são atualizadas mensalmente sobre o uso principal do ElastiCache em dados categóricos, como:
 - a. Tipos de nó usados e seus atributos: instâncias padrão versus instâncias otimizadas para memória, instâncias sob demanda versus instâncias reservadas, regiões e zonas de disponibilidade
 - b. Tipos de ambiente: gratuito, desenvolvimento, teste e produção
 - c. Estratégias de backup, armazenamento e retenção
 - d. Transferência de dados dentro de uma região e entre regiões
 - e. Instâncias em execução no Amazon Outposts
 2. O CCoE consiste em uma equipe multifuncional com representação não exclusiva das equipes de engenharia de software, gerenciamento de dados, produtos, finanças e liderança em sua organização.

[Recursos]:

- [Criar um Centro de Excelência da Nuvem](#)
- [Preço do Amazon ElastiCache](#)

- [Obrigatório] Use etiquetas de alocação de custos para monitorar os custos em um baixo nível de granularidade. Use o Gerenciamento de Custos da AWS para visualizar, entender e gerenciar seus custos e uso da AWS ao longo do tempo.
 1. Use etiquetas para organizar seus recursos e etiquetas de alocação de custos para monitorar os custos da AWS em detalhes. Depois de ativar as etiquetas de alocação de custos, a AWS as utiliza para organizar seus custos de recursos no relatório de alocação de custos para facilitar a categorização e o controle dos seus custos na AWS. A AWS fornece dois tipos de etiquetas de alocação de custos: etiquetas geradas pela AWS e etiquetas definidas por usuário. A AWS define, cria e aplica as etiquetas geradas pela AWS para você, enquanto você define, cria e aplica as etiquetas definidas por usuário. É necessário ativar os dois tipos de etiquetas separadamente para que elas possam ser exibidas no Gerenciamento de Custos ou em um relatório de alocação de custos.
 2. Use tags de alocação de custos para organizar sua fatura da AWS para refletir sua própria estrutura de custos. Quando você adiciona etiquetas de alocação de custos aos seus recursos no Amazon ElastiCache, pode acompanhar os custos agrupando as despesas nas suas faturas por valores de etiqueta de recursos. Considere combinar etiquetas para monitorar os custos com um maior nível de detalhes.

[Recursos]:

- [Usar etiquetas de alocação de custos da AWS](#)
- [Monitorar os custos com etiquetas de alocação de custos](#)
- [AWS Cost Explorer](#)
- [Ideal] Conecte o custo do ElastiCache às métricas que abrangem toda a organização.
 1. Considere as métricas de negócios e as métricas operacionais, como latência: quais conceitos em seu modelo de negócios são compreensíveis em todas as funções? As métricas precisam ser compreensíveis pelo maior número possível de funções na organização.
 2. Exemplos: usuários atendidos simultaneamente, latência máxima e média por operação e usuário, pontuações de engajamento do usuário, taxas de retorno do usuário/semana, duração da sessão/usuário, taxa de abandono, taxa de acertos de cache e chaves monitoradas.

[Recursos]:

- [Monitorar o uso com métricas do CloudWatch](#)
- [Bom] Mantenha uma visibilidade arquitetônica e operacional atualizada sobre métricas e custos em toda a workload que usa o ElastiCache.

1. Entenda todo o seu ecossistema de soluções. O ElastiCache tende a fazer parte de um ecossistema completo de serviços da AWS em seu conjunto de tecnologias, desde clientes até API Gateway, Redshift e QuickSight para ferramentas de geração de relatórios (por exemplo).
2. Mapeie os componentes da sua solução, desde clientes, conexões, segurança, operações em memória, armazenamento, automação de recursos, acesso e gerenciamento de dados, em seu diagrama de arquitetura. Cada camada se conecta a toda a solução e tem suas próprias necessidades e recursos que aumentam e/ou ajudam a gerenciar o custo geral.
3. Seu diagrama deve incluir o uso de políticas de computação, rede, armazenamento, ciclo de vida, coleta de métricas, bem como os elementos operacionais e funcionais do ElastiCache da sua aplicação.
4. É provável que os requisitos de uma workload evoluam com o tempo e é essencial que você continue a manter e documentar sua compreensão dos componentes subjacentes, bem como de seus principais objetivos funcionais, a fim de manter a proatividade no gerenciamento de custos da workload.
5. O suporte executivo para visibilidade, responsabilização, priorização e recursos é crucial para que você tenha uma estratégia eficaz de gerenciamento de custos para o ElastiCache.

CUSTOS 2: Como você usa ferramentas de monitoramento contínuo para ajudar a otimizar os custos associados aos seus recursos do ElastiCache?

Introdução: você precisa buscar um equilíbrio adequado entre os custos do ElastiCache e as métricas de performance da aplicação. O Amazon CloudWatch fornece visibilidade das principais métricas operacionais que podem te ajudar a avaliar se seus recursos do ElastiCache estão sendo usados em excesso ou subutilizados em relação às suas necessidades. Do ponto de vista da otimização de custos, você precisa entender quando há excesso de provisionamento e ser capaz de desenvolver mecanismos apropriados para redimensionar seus recursos do ElastiCache, ao mesmo tempo em que mantém suas necessidades de operação, disponibilidade, resiliência e performance.

Benefício: em um estado ideal, você terá provisionado recursos suficientes para atender às necessidades operacionais de sua workload e não terá recursos subutilizados que possam levar a um estado de custo abaixo do ideal. Você precisa ser capaz de identificar e evitar a operação de recursos superdimensionados do ElastiCache por longos períodos.

- [Obrigatório] Use o CloudWatch para monitorar seus clusters do ElastiCache e analisar como essas métricas se relacionam com seus painéis do Explorador de custos da AWS.

1. O ElastiCache fornece métricas em nível de host (por exemplo, uso da CPU) e métricas que são específicas do software do mecanismo de cache (por exemplo, obtenções e ausências no cache). Essas métricas são medidas e publicadas para cada nó de cache em intervalos de 60 segundos.
2. As métricas de performance do ElastiCache (CPUUtilization, EngineUtilization, SwapUsage, CurrConnections e Evictions) podem indicar que você precisa aumentar/diminuir a escala verticalmente (usar tipos de nó de cache maiores/menores) ou aumentar/diminuir a escala horizontalmente (adicionar mais/menos fragmentos). Entenda as implicações de custo das decisões de ajuste de escala criando uma matriz de playbook que estima o custo adicional e os períodos mínimo e máximo necessários para atingir os limites de performance de sua aplicação.

[Recursos]:

- [Monitorar o uso com métricas do CloudWatch](#)
 - [Quais métricas devo monitorar?](#)
 - [Preço do Amazon ElastiCache](#)
- [Obrigatório] Entenda e documente sua estratégia de backup e as implicações de custo.
 1. Com o ElastiCache, os backups são armazenados no Amazon S3, que fornece armazenamento durável. Você precisa entender as implicações de custo em relação à sua capacidade de se recuperar de falhas.
 2. Ative backups automáticos que vão excluir os arquivos de backup que ultrapassaram o limite de retenção.

[Recursos]:

- [Programação de backups automáticos](#)
 - [Preços do Amazon Simple Storage Service](#)
- [Ideal] Use nós reservados para suas instâncias como uma estratégia deliberada para gerenciar custos de workloads que sejam bem compreendidas e documentadas. Para nós reservados, é necessário pagar uma taxa adiantada que depende do tipo de nó e da duração da reserva, um ou três anos. Essa cobrança é muito menor do que a cobrança de uso por hora dos nós sob demanda.
 1. Talvez seja necessário operar os clusters do ElastiCache usando nós sob demanda até reunir dados suficientes para estimar os requisitos da instância reservada. Planeje e documente os recursos necessários para atender às suas necessidades e compare os custos esperados em todos os tipos de instância (sob demanda versus reservada)

2. Avalie regularmente os novos tipos de nó de cache disponíveis e avalie se faz sentido, do ponto de vista das métricas de operação e custo, migrar sua frota de instâncias para novos tipos de nó de cache.

CUSTOS 3: Você deve usar um tipo de instância que ofereça suporte à hierarquização de dados? Quais são as vantagens de uma hierarquização de dados? Quando não usar instâncias de hierarquização de dados?

Introdução: selecionar o tipo de instância apropriado pode afetar não só a performance e o serviço, mas também as finanças. Cada tipo de instância tem custos diferentes. Selecionar um ou alguns tipos de instâncias grandes que possam acomodar todas as necessidades de armazenamento em memória pode ser uma decisão natural. No entanto, isso pode afetar significativamente os custos à medida que o projeto amadurece. Garantir que o tipo de instância correto seja selecionado exige uma avaliação periódica do tempo de inatividade do objeto do ElastiCache.

Benefício: você deve ter uma compreensão clara de como os vários tipos de instância afetam seus custos no presente e no futuro. Mudanças marginais ou periódicas na workload não devem provocar mudanças desproporcionais nos custos. Se a workload permitir, os tipos de instância compatíveis com a hierarquização de dados oferecem um preço melhor por armazenamento disponível. Devido à disponibilidade por instância de armazenamento SSD, as instâncias com hierarquização de dados oferecem suporte a uma capacidade total de dados muito maior por instância.

- [Obrigatório] Entenda as limitações das instâncias com hierarquização de dados.
 1. Disponível somente para clusters do ElastiCache para Redis.
 2. Somente alguns tipos de instância oferecem suporte à hierarquização de dados.
 3. Somente o ElastiCache para Redis versão 6.2 e superior é compatível.
 4. Itens grandes não são trocados para SSD. Objetos com mais de 128 MiB são mantidos na memória.

[Recursos]:

- [Hierarquização de dados](#)
- [Preço do Amazon ElastiCache](#)
- [Obrigatório] Entenda qual porcentagem do seu banco de dados é acessada regularmente pela workload.

1. As instâncias com hierarquização de dados são ideais para workloads que geralmente acessam uma pequena parte do conjunto geral de dados, mas que ainda exigem acesso rápido aos dados restantes. Em outras palavras, a proporção de dados quentes para mornos é de cerca de 20:80.
 2. Desenvolva o monitoramento do tempo ocioso do objeto no cluster.
 3. Grandes implementações de mais de 500 Gb de dados são boas candidatas.
- [Obrigatório] Entenda que as instâncias com hierarquização de dados não são opcionais para determinadas workloads.
 1. Há um pequeno custo de performance para acessar objetos usados com menos frequência, pois eles são trocados para um SSD local. Se sua aplicação for sensível ao tempo de resposta, teste o impacto na workload.
 2. Não é adequado para caches que armazenam principalmente objetos grandes com mais de 128 MiB.

[Recursos]:

- [Limitações](#)
- [Ideal] Os tipos de instância reservada oferecem suporte à hierarquização de dados. Isso garante o menor custo em termos de quantidade de armazenamento de dados por instância.
 1. Talvez seja necessário operar seus clusters do ElastiCache usando instâncias sem hierarquização de dados até ter uma melhor compreensão de seus requisitos.
 2. Analise seu padrão de uso de dados de clusters do ElastiCache.
 3. Crie um trabalho automatizado que colete periodicamente o tempo ocioso do objeto.
 4. Se você perceber que uma grande porcentagem (cerca de 80%) dos objetos fica ociosa por um período considerado apropriado para sua workload, documente as descobertas e sugira a migração do cluster para instâncias compatíveis com a hierarquização de dados.
 5. Avalie regularmente os novos tipos de nó de cache disponíveis e avalie se faz sentido, do ponto de vista das métricas de operação e custo, migrar sua frota de instâncias para novos tipos de nó de cache.

[Recursos]:

- [OBJECT IDLETIME](#)
- [Preço do Amazon ElastiCache](#)

Etapas comuns de solução de problemas e melhores práticas

Tópicos

- [Problemas de conectividade](#)
- [Erros do cliente Redis](#)
- [Solução de problemas de alta latência no Serverless ElastiCache](#)
- [Solução de problemas de limitação no Serverless ElastiCache](#)
- [Related Topics](#)

Problemas de conectividade

Se você não conseguir se conectar ao ElastiCache cache, considere uma das seguintes opções:

1. Usando o TLS: se você estiver com uma conexão interrompida ao tentar se conectar ao seu ElastiCache endpoint, talvez não esteja usando o TLS no seu cliente. Se você estiver usando o ElastiCache Serverless, a criptografia em trânsito estará sempre ativada. Certifique-se de que seu cliente esteja usando o TLS para se conectar ao cache. Saiba mais sobre como se conectar a um cache habilitado para TLS [aqui](#).
2. VPC: ElastiCache os caches só podem ser acessados de dentro de uma VPC. Certifique-se de que a instância do EC2 a partir da qual você está acessando o cache e o ElastiCache cache sejam criados na mesma VPC. Como alternativa, você deve ativar o [emparelhamento de VPC](#) entre a VPC em que sua instância do EC2 reside e a VPC em que você está criando seu cache.
3. Grupos de segurança: ElastiCache usa grupos de segurança para controlar o acesso ao seu cache. Considere o seguinte:
 - a. Certifique-se de que o grupo de segurança usado pelo seu ElastiCache cache permita acesso de entrada a ele a partir da sua instância do EC2. 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 do Redis. Saiba mais sobre como configurar o acesso à porta [aqui](#).

Erros do cliente Redis

ElastiCache O Serverless só pode ser acessado usando clientes Redis que oferecem suporte ao protocolo do modo de cluster Redis. Clusters autoprojetoados podem ser acessados a partir de clientes Redis em qualquer modo, dependendo da configuração do cluster.

Se você estiver enfrentando erros do Redis em seu cliente, considere o seguinte:

1. Modo de cluster: se você estiver enfrentando erros de CROSSLOT ou erros com o comando [SELECT](#) Redis, talvez esteja tentando acessar um cache habilitado para o modo de cluster com um cliente Redis que não oferece suporte ao protocolo Redis Cluster. ElastiCache O Serverless só oferece suporte a clientes Redis que oferecem suporte ao protocolo de cluster Redis. Se você quiser usar o Redis no “Modo de cluster desativado” (CMD), deverá criar seu próprio cluster.
2. Erros de CROSSLOT: 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 clientes 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.

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 cliente 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 nas mesmas AZs do seu cache:** você pode observar uma maior latência do lado do cliente se seu aplicativo não for implantado nas mesmas AZs do seu cache. Ao criar um cache sem servidor, você pode fornecer as sub-redes de onde seu aplicativo acessará o cache, e o Serverless ElastiCache cria VPC Endpoints nessas sub-redes. Certifique-se de que seu aplicativo seja implantado nas mesmas 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 conexão TCP habilitada para TLS usando o protocolo RESP. Iniciar a conexão (incluindo a autenticação da conexão, se configurada) leva tempo, então a latência da primeira solicitação é maior do que a normal. Solicitações em uma conexão já inicializada oferecem ElastiCache baixa latência consistente. Por esse motivo, você deve considerar o uso do pool de conexões ou a reutilização das conexões Redis existentes.
- **Velocidade de escalonamento:** o ElastiCache Serverless é escalado automaticamente à medida que sua taxa de solicitações aumenta. Um grande aumento repentino na taxa de solicitações, mais rápido do que a velocidade na qual o ElastiCache Serverless é escalado, pode resultar em latência elevada por algum tempo. ElastiCache Normalmente, o Serverless pode aumentar rapidamente a taxa de solicitações suportadas, levando de 10 a 12 minutos para dobrar a taxa de solicitações.
- **Inspecione comandos de longa execução:** alguns comandos do Redis, 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: no ElastiCache Redis, uma distribuição desigual de chaves ou solicitações por slot pode resultar em um hot slot, o que pode resultar em latência elevada. ElastiCache O Serverless suporta até 30.000 ECPUS/segundo (90.000 ECPUS/segundo ao usar Read from Replica) em um único slot, em uma carga de trabalho que executa comandos SET/GET simples. Recomendamos avaliar a distribuição da chave e da solicitação em todos os slots e garantir uma distribuição uniforme se a taxa de solicitação exceder esse limite.

Solução de problemas de limitação no Serverless ElastiCache

Em arquiteturas orientadas a serviços e sistemas distribuídos, a limitação da taxa na qual as chamadas de API são processadas por vários componentes do serviço é chamada de controle de utilização. Isso suaviza os picos, controla as incompatibilidades na produtividade dos componentes e permite recuperações mais previsíveis quando há um evento operacional inesperado. ElastiCache O Serverless foi projetado para esses tipos de arquiteturas, e a maioria dos clientes 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: no ElastiCache Redis, 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 solicitação para um único slot exceder 30.000 ECPUs por segundo, em uma carga de trabalho que executa comandos SET/GET simples.
- Leia da réplica: se seu aplicativo permitir, considere usar o recurso “Ler da réplica”. A maioria dos clientes 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 por segundo em um único slot, para cargas de trabalho com comandos SET/GET simples.

Related Topics

- [Etapas adicionais de solução de](#)
- [the section called “Práticas recomendadas e estratégias de armazenamento em cache”](#)

Etapas adicionais de solução de

Os itens a seguir devem ser verificados durante a solução de problemas persistentes de conectividade com ElastiCache:

Tópicos

- [Grupos de segurança](#)
- [Network ACLs](#)
- [Tabelas de rotas](#)
- [Resolução do DNS](#)
- [Identificação de problemas com o diagnóstico do lado do servidor](#)
- [Validação da conectividade de rede](#)
- [Limites relacionados à rede](#)
- [Uso da CPU](#)
- [Conexões sendo encerradas do lado do servidor](#)
- [Solução de problemas no lado do cliente para instâncias do Amazon EC2](#)
- [Dissecação do tempo necessário para concluir uma única solicitação](#)

Grupos de segurança

Grupos de segurança são firewalls virtuais que protegem seu ElastiCache cliente (instância EC2, AWS Lambda função, contêiner Amazon ECS etc.) e cache. ElastiCache Os grupos de segurança têm estado, o que significa que depois que o tráfego de entrada ou saída é permitido, as respostas

para esse tráfego serão automaticamente autorizadas no contexto desse grupo de segurança específico.

O recurso stateful (com estado) requer que o grupo de segurança mantenha o controle de todas as conexões autorizadas e há um limite para conexões controladas. Se o limite for atingido, as novas conexões apresentarão falha. Consulte a seção de solução de problemas para obter ajuda sobre como identificar se os limites foram atingidos 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 saída TCP na ElastiCache porta da origem e o tráfego de entrada na mesma porta para. ElastiCache A porta padrão é 11211 para o Memcached e 6379 para o Redis. 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](#).

Network ACLs

As listas de controle de acesso à rede (ACLs) são regras sem estado. O tráfego deve ser permitido em ambas as direções (entrada e saída) para ter êxito. As ACLs de rede são atribuídas a sub-redes, não a recursos específicos. É possível ter a mesma ACL atribuída ElastiCache e ao recurso do cliente, especialmente se eles estiverem na mesma sub-rede.

Por padrão, as ACLs de rede permitem todo o tráfego. No entanto, é possível personalizá-las para negar ou permitir tráfego. Além disso, a avaliação das regras da ACL é sequencial, o que significa que a regra com o menor número correspondente ao tráfego irá permiti-lo ou negá-lo. A configuração mínima para permitir o tráfego do Redis é:

ACL de rede do lado do cliente:

- Regras de entrada:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Type: Custom TCP Rule;
- Protocolo: TCP
- Port Range: 1024-65535

- 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;
- Type: Custom TCP Rule;
- Protocolo: TCP
- Port Range: 6379
- Fonte: 0.0.0.0/0 (ou as sub-redes do cluster). ElastiCache Lembre-se de que o uso de IPs específicos pode criar problemas em caso de failover (ou escalonamento do cluster).
- Allow/Deny: Allow

ElastiCache ACL de rede:

- Regras de entrada:
- Número da regra: preferencialmente inferior a qualquer regra de negação;
- Type: Custom TCP Rule;
- 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;
- Type: Custom TCP Rule;
- 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 de IPs específicos pode criar problemas em caso de failover (ou escalonamento do cluster).
- Allow/Deny: Allow

Para obter mais informações, consulte [ACLs de rede](#).

Tabelas de rotas

Da mesma forma que as ACLs de rede, cada sub-rede pode ter tabelas de rota diferentes. Se os clientes e o ElastiCache cluster estiverem em sub-redes diferentes, certifique-se de que suas tabelas de rotas permitam que eles entrem em contato uns com os outros.

Ambientes mais complexos, envolvendo várias VPCs, roteamento dinâmico ou firewalls de rede, podem se tornar difíceis para a solução de problemas. Consulte [Validação da conectividade de rede](#) para confirmar se as configurações de rede são apropriadas.

Resolução do DNS

ElastiCache fornece os endpoints do serviço com base em nomes de DNS. Os endpoints disponíveis são os endpoints Configuration, Primary, Reader, e Node. Para obter mais informações, consulte [Encontrar endpoints de conexão](#).

Em caso de failover ou modificação de cluster, o endereço associado ao nome do endpoint pode mudar e será atualizado automaticamente.

As configurações personalizadas de DNS (ou seja, não usar o serviço VPC DNS) podem não estar cientes ElastiCache dos nomes DNS 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 de nomes por meio do serviço de DNS da VPC:

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Identificação de problemas com o diagnóstico do lado do servidor

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:

- **Uso da CPU:** o Redis é uma aplicação multi-threaded. No entanto, a execução de cada comando ocorre em um único thread (principal). Por esse motivo, ElastiCache fornece as métricas `CPUUtilization` e `EngineCPUUtilization`. `CPUUtilization` fornece a utilização da CPU dedicada ao processo Redis e o uso em todas as vCPUs. Os nós com mais de uma vCPU geralmente têm valores diferentes para `CPUUtilization` e `EngineCPUUtilization`, sendo o segundo comumente maior. Um `EngineCPUUtilization` alto pode ser causado por um número elevado de solicitações ou operações complexas que levam uma quantidade significativa de tempo da CPU para ser concluída. Você pode identificar ambos com o seguinte:
 - **Número elevado de solicitações:** verifique se há aumentos em outras métricas que correspondam ao padrão `EngineCPUUtilization`. As métricas úteis são:
 - **CacheHits e CacheMisses:** o número de solicitações ou solicitações bem-sucedidas que não encontraram um item válido no cache. Se a proporção de erros em comparação com acertos for alta, a aplicação está desperdiçando tempo e recursos com solicitações infrutíferas.
 - **SetTypeCmds e GetTypeCmds:** Essas métricas correlacionadas com `EngineCPUUtilization` podem ajudar a entender se a carga é significativamente maior para solicitações de gravação, medida por `SetTypeCmds`, ou leituras, medido por `GetTypeCmds`. Se a carga for predominantemente de leituras, o uso de várias réplicas de leitura pode equilibrar as solicitações em vários nós e poupar o primário para gravações. Em clusters 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 certos, a biblioteca será capaz de descobrir automaticamente a topologia do cluster, os nós de réplica, habilitar as operações de leitura através do comando `READONLY` do Redis e enviar as solicitações de leitura para as réplicas.
 - **Número elevado de conexões:**
 - **CurrConnections e NewConnections:** `CurrConnection` é o número de conexões estabelecidas no momento da coleta de pontos de dados, enquanto `NewConnections` mostra quantas conexões foram criadas no período.

Criar e manipular conexões implica em sobrecarga significativa da CPU. Além disso, o handshake de três vias de TCP necessário para criar novas conexões afetará negativamente os tempos de resposta gerais.

Um 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 Redis fornece performance ideal com um pequeno número de `currConnections`. Manter `currConnection` na ordem de dezenas ou centenas minimiza o uso de recursos para suportar conexões individuais, como buffers de cliente e ciclos de CPU para servir a conexão.

- Throughput da rede
 - Determine a largura de banda: 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, aplicações com alta taxa de pequenas solicitações tendem a afetar mais o uso da CPU do que o throughput da rede, enquanto chaves maiores causarão maior utilização da rede. Por esse motivo, é aconselhável testar os nós com a workload real para uma melhor compreensão dos limites.

A simulação da carga da aplicação forneceria resultados mais precisos. No entanto, as ferramentas de benchmark podem dar uma boa ideia dos limites.

- Para casos em que as solicitações são predominantemente leituras, o uso de réplicas para operações de leitura aliviará a carga no nó primário. Se o caso de uso for predominantemente gravações, o uso de muitas réplicas amplificará o uso da rede. Para cada byte gravado no nó primário, N bytes serão enviados para as réplicas, sendo N o número de réplicas. A melhor prática para cargas de trabalho intensivas de gravação é usar o Redis com o modo de cluster habilitado ElastiCache para que as gravações possam ser balanceadas em vários fragmentos ou escaladas 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 ter mais informações, consulte [Limites relacionados à rede](#).

- Comandos complexos: os comandos do Redis 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 keypace procurando por um determinado padrão e bloqueia a execução de outros comandos durante sua execução. O Redis usa a notação "Big O" para descrever sua complexidade de comandos.

O comando Keys tem complexidade de tempo $O(N)$, sendo N o número de chaves no banco de dados. Portanto, quanto maior o número de chaves, mais lento será o comando. KEYS pode causar problemas de maneiras diferentes: se nenhum padrão de pesquisa for usado, o comando retornará todos os nomes de chaves disponíveis. Em bancos de dados com milhares ou milhões de itens, uma enorme saída será criada e inundará os buffers de rede.

Se um padrão de pesquisa for usado, somente as chaves correspondentes ao padrão retornarão ao cliente. No entanto, o mecanismo ainda varrerá todo o keypace 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 keypace e limita as iterações em um número específico de itens, evitando bloqueios prolongados no mecanismo.

A varredura tem o parâmetro COUNT, usado para definir o tamanho dos blocos de iteração. O valor padrão é 10 (10 itens por iteração).

Dependendo do número de itens no banco de dados, blocos com pequenos valores de COUNT irão exigir mais iterações para concluir uma verificação completa, e valores maiores manterão o mecanismo ocupado por mais tempo em cada iteração. Enquanto pequenos valores de contagem farão SCAN mais lento em grandes bancos de dados, valores maiores podem causar os mesmos problemas mencionados para KEYS.

Como exemplo, a execução do comando SCAN com valor de contagem de 10 vai requer 100.000 repetições em um banco de dados com 1 milhão de chaves. Se o tempo médio de ida e volta da rede for de 0,5 milissegundos, aproximadamente 50.000 milissegundos (50 segundos) serão gastos transferindo solicitações.

Por outro lado, se o valor da contagem fosse 100.000, uma única iteração seria necessária e apenas 0,5 ms seriam gastos transferindo-a. No entanto, o mecanismo seria totalmente bloqueado para outras operações até que o comando terminasse de varrer todo o keyspace.

Além de KEYS, vários outros comandos são potencialmente prejudiciais se não forem usados corretamente. Para ver uma lista de todos os comandos e suas respectivas complexidades de tempo, acesse <https://redis.io/commands>.

Exemplos de possíveis problemas:

- **Scripts Lua:** o Redis fornece um interpretador Lua incorporado, permitindo a execução de scripts no lado do servidor. Os scripts Lua no Redis são executados no nível do mecanismo e são atômicos por definição, o que significa que nenhum outro comando ou script terá execução permitida enquanto um script estiver em execução. Os scripts Lua fornecem a possibilidade de executar vários comandos, algoritmos de tomada de decisão, análise de dados e outros diretamente no mecanismo do Redis. 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. [Transações](#) são a alternativa para garantir a consistência de múltiplas modificações de chaves relacionadas no Redis. Uma transação permite a execução de um bloco de comandos, observando as chaves existentes para modificações. Se qualquer uma das chaves observadas mudar antes da conclusão da transação, todas as modificações serão descartadas.
- **Exclusão em massa de itens:** o comando DEL aceita vários parâmetros, que são os nomes das chaves a serem excluídas. As operações de exclusão são síncronas e gastarão um tempo significativo da CPU se a lista de parâmetros for grande ou contiver uma grande lista, conjunto, conjunto ordenado ou hash (estruturas de dados contendo vários subitens). Em outras palavras, até mesmo a exclusão de uma única chave pode levar um tempo significativo se tiver muitos elementos. A alternativa a DEL é UNLINK, que é um comando assíncrono disponível desde o Redis 4. UNLINK deve ter preferência sobre DEL sempre que possível. A partir do ElastiCache Redis 6x, o `lazyfree-lazy-user-del` parâmetro faz com que o DEL comando se comporte como UNLINK quando ativado. Para mais informações, consulte [Alterações de parâmetro do Redis 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 fornece muitos outros comandos que funcionam de forma semelhante. Como exemplos, MSET e MGET permitem a inserção ou recuperação de várias chaves de string ao mesmo tempo. Seu uso pode ser benéfico para reduzir a latência de rede inerente a vários comandos individuais SET ou GET. No entanto, uma extensa lista de parâmetros afetará o uso da CPU.

Embora a utilização da CPU por si só não seja a causa de problemas de conectividade, gastar muito tempo para processar um único ou poucos comandos através de várias chaves pode causar falha de outras solicitações e aumentar a utilização geral da CPU.

O número de chaves e seu tamanho afetarão a complexidade do comando e, 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 atuando sobre vários tipos de dados: o Redis também fornece comandos que atuam em uma ou várias chaves, independentemente do tipo de dados. ElastiCache for Redis fornece a métrica KeyBasedCmds para monitorar esses comandos. Essa métrica soma a execução dos seguintes comandos no período selecionado:
 - Complexidade de $O(N)$:
 - KEYS
 - $O(1)$
 - EXISTS
 - OBJECT
 - PTTL
 - RANDOMKEY
 - TTL
 - TYPE
 - EXPIRE
 - EXPIREAT
 - MOVE

- PERSIST
- PEXPIRE
- PEXPIREAT
- UNLINK ($O(N)$) para recuperar memória. No entanto, a tarefa de recuperação de memória ocorre em um thread separado e não bloqueia o mecanismo
- Diferentes tempos de complexidade dependendo do tipo de dados:
 - DEL
 - DUMP
 - RENAME é considerado um comando com complexidade $O(1)$, mas executa DEL internamente. O tempo de execução varia de acordo com o tamanho da chave renomeada.
 - RENAMENX
 - RESTORE
 - SORT
- Hashes grandes: hash é um tipo de dado que permite uma única chave com vários subitens de valor-chave. Cada hash pode armazenar 4.294.967.295 itens e operações em hashes grandes podem se tornar caras. Da mesma forma que KEYS, hashes têm o comando HKEYS com complexidade de tempo $O(N)$, N sendo o número de itens no hash. HSCAN deve ter preferência sobre HKEYS para evitar comandos de longa execução. HDEL, HGETALL, HMGET, HMSET e HVALS são comandos que devem ser usados com cautela em hashes grandes.
- Outras estruturas grandes de dados: além de hashes, outras estruturas de dados podem ser intensas em CPU. Conjuntos, listas, conjuntos classificados e HyperLogLogs também podem levar um tempo significativo para serem manipulados, dependendo do tamanho e dos comandos usados. Para obter mais informações sobre esses comandos, consulte <https://redis.io/commands>.

Validação da conectividade de rede

Depois de analisar as configurações de rede relacionadas à resolução de DNS, grupos de segurança, ACLs de rede e tabelas de rota, a conectividade pode ser validada com o VPC Reachability Analyzer e as ferramentas do sistema.

O Reachability Analyzer testará a conectividade de rede e confirmará se todos os requisitos e permissões estão satisfeitos. Para os testes abaixo, você precisará do ENI ID (Elastic Network Interface Identification) de um dos ElastiCache nós disponíveis em sua VPC. É possível fazer isso da seguinte maneira:

1. Acesse <https://console.aws.amazon.com/ec2/v2/home?#NIC:>
2. Filtre a lista de interfaces pelo nome do ElastiCache cluster ou pelo endereço IP obtido das validações de DNS anteriores.
3. Anote ou salve o ID da ENI. 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. Crie um caminho de análise em [https://console.aws.amazon.com/vpc/home? # ReachabilityAnalyzer](https://console.aws.amazon.com/vpc/home?#ReachabilityAnalyzer) e escolha as seguintes opções:
 - Tipo de fonte: Escolha a instância se seu ElastiCache cliente for executado em uma instância do Amazon EC2 ou interface de rede se usar outro serviço, como AWS Fargate Amazon ECS com rede awsvpc, etc.) AWS Lambda, e o respectivo ID do recurso (instância EC2 ou ID ENI);
 - Destination Type (Tipo de destino): escolha Network Interface (Interface de rede) e selecione a ElastiCache ENI (ENI do ElastiCache) na lista.
 - Porta de destino: especifique 6379 ElastiCache para Redis 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 conectividade TCP na porta de ElastiCache serviço: No Amazon Linux, Nping está disponível no pacote nmap e pode testar a conectividade TCP 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 com nping falharem e os testes do VPC Reachability Analyzer forem aprovados, peça ao administrador do sistema para revisar possíveis regras de firewall baseadas em host, regras de roteamento assimétricas ou qualquer outra restrição possível no nível do sistema operacional.

No ElastiCache console, verifique se a criptografia em trânsito está habilitada nos detalhes ElastiCache do seu cluster. Se a criptografia em trânsito estiver habilitada, confirme se a sessão de TLS pode ser estabelecida com o seguinte comando:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Uma saída extensa é esperada se a conexão e a negociação de TLS forem bem-sucedidas. Verifique o código de retorno disponível na última linha, o valor deve ser de 0 (ok). Se o openssl retornar algo diferente, verifique o motivo do erro em <https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS>.

Se todos os testes de infraestrutura e sistema operacional 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 é compatível com TLS/SSL e ElastiCache tem a criptografia em trânsito ativada;
- A aplicação suporta TLS/SSL, mas não possui os sinalizadores de configuração corretos ou autoridades de certificação confiáveis;

Limites relacionados à rede

- Número máximo de conexões: há limites rígidos para conexões simultâneas. Cada 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](#) 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 períodos prolongados, eles provavelmente serão seguidos por atraso de replicação, aumento de bytes usados para cache, queda de memória livre, alto swap e uso da CPU. As instâncias do Amazon EC2 também têm limites de rede que podem ser rastreados por meio de [Métricas do driver ENA](#). Instâncias do Linux com suporte de rede aprimorado e drivers ENA 2.2.10 ou mais recentes podem revisar os contadores de limite com o comando:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

Uso da CPU

A métrica de uso da CPU é o ponto de partida da investigação, e os itens a seguir podem ajudar a reduzir possíveis problemas ElastiCache colaterais:

- **Redis 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 19:26:07 UTC, levou 4,8 segundos (4,823 ms) para ser concluído e foi causado pelo comando KEYS solicitado pelo cliente 1.2.3.4.

No Linux, o carimbo de data/hora pode ser convertido com a data do comando:

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Com Python:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

Ou no Windows com PowerShell:

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
DateTime           : 12/26/2020 7:26:07 PM
UtcDateTime        : 12/26/2020 7:26:07 PM
LocalDateTime      : 12/26/2020 2:26:07 PM
Date               : 12/26/2020 12:00:00 AM
Day               : 26
DayOfWeek          : Saturday
DayOfYear         : 361
Hour              : 19
Millisecond        : 0
Minute            : 26
Month             : 12
Offset            : 00:00:00Ticks           : 637446075670000000
UtcTicks          : 637446075670000000
TimeOfDay         : 19:26:07
Year              : 2020
```

Muitos comandos lentos em um curto período de tempo (mesmo minuto ou menos) são motivo de preocupação. Revise a natureza dos comandos e como eles podem ser otimizados (consulte exemplos anteriores). Se comandos com complexidade de tempo $O(1)$ são frequentemente relatados, verifique os outros fatores para o alto uso da CPU mencionado anteriormente.

- **Métricas de latência:** ElastiCache for Redis 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`;
 - **GetTypeCmdsLatency:** Leia comandos, independentemente do tipo de dados;
 - **HashBasedCmdsLatency:** `hexists`, `hget`, `hgetall`, `hkeys`, `hlen`, `hmget`, `hvals`, `hstrlen`, `hdel`, `hincrby`, `hincrbyfloat`, `hset`, `hsetnx`;
 - **HyperLogLogBasedCmdsLatency:** `pfselftest`, `pfcount`, `pfdebug`, `pfadd`, `pfmerge`;
 - **KeyBasedCmdsLatency:** Comandos que podem atuar em diferentes tipos de dados: `dump`, `existskeys`, `object`, `pttl`, `randomkey`, `ttdel`, `ttl`, `type`, `del`, `expire`, `expireat`, `move`, `persist`, `pexpire`;
 - **ListBasedCmdsLatency:** `lindex`, `len`, `lrange`, `lpop`, `lpush`, `lpushx`, `lrem`, `ltrim`, `rop`, `roplpush`, `rpush`, `rpushx`;
 - **PubSubBasedCmdsLatency:** `psubscribe`, `publike`, `pubsub`, `unsubscribe`, `subscribe`, `unsubscribe`;
 - **SetBasedCmdsLatency:** `scard`, `sdiff`, `sinter`, `sismember`, `smembers`, `srandmember`, `sunion`, `sadd`, `sdiffstore`, `sinterstore`, `smove`, `spop`, `srem`, `sunionstore`;
 - **SetTypeCmdsLatency:** Escreva comandos, independentemente do tipo de dados;
 - **SortedSetBasedCmdsLatency:** `zcard`, `zcount`, `zrange`, `zrangebyscore`, `zrank`, `zrevrange`, `zrevrangebyscore`, `zrevrank`, `zscore`, `zrangebylex`, `zrevrangebylex`, `zlexcount`, `zadd`, `zincrby`, `zinterstore`, `zrem`, `zremrangebyrank`, `zremrangebyscore`, `zunionstore`, `zremrangebylex`, `zpopmax`, `zpopmin`, `bzpopmin`, `bzpopmax`;

- **StringBasedCmdsLatency:** `bitcount`, `get`, `getbit`, `getrange`, `mget`, `strlen`, `substr`, `bitpos`, `append`, `bitop`, `bitfield`, `decr`, `decrby`, `getset`, `incr`, `incrby`, `incrbyfloat`, `mset`, `msetnx`, `psetex`, `set`, `setbit`, `setex`, `setnx`, `setrange`;
- **StreamBasedCmdsLatency:** `xrange`, `xrevrange`, `xlen`, `xread`, `xpending`, `xinfo`, `xadd`, `xgroup`, `readgroup`, `xack`, `xclaim`, `xdel`, `xtrim`, `xsetid`;
- **Comandos de runtime do Redis:**
 - `info commandstats`: fornece uma lista de comandos executados desde que o mecanismo do Redis foi iniciado, seu número de execuções cumulativas, tempo total de execução e tempo médio de execução por comando;
 - `client list`: fornece uma lista de clientes atualmente conectados e informações relevantes como o uso de buffers, último comando executado, etc.;
- **Backup e replicação:** ElastiCache para versões do Redis anteriores à 2.8.22, use um processo bifurcado para criar backups e processar sincronizações completas com as réplicas. Esse método pode incorrer em sobrecarga de memória significativa para casos de uso intensivo de gravação.

A partir do ElastiCache Redis 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 utilização da CPU, levar a tempos de resposta mais altos e, conseqüentemente, levar a estouro de limites de tempo do cliente durante a execução. Sempre verifique se as falhas do cliente acontecem durante a janela de backup ou a métrica `SaveInProgress` foi 1 no período. É aconselhável agendar a janela de backup para períodos de baixa utilização para minimizar a possibilidade de problemas com clientes ou falhas de backup.

Conexões sendo encerradas do lado do servidor

O padrão ElastiCache para a configuração do Redis 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 conseqüência 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 qualquer solicitação no período, o 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. O padrão ElastiCache para configurações do Redis não restringe 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 for Redis 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 no lado do cliente para instâncias do Amazon EC2

A carga e a capacidade de resposta do lado do cliente também podem afetar as solicitações de ElastiCache. Os limites de instância e sistema operacional do EC2 precisam ser cuidadosamente revisados ao solucionar problemas de conectividade intermitente ou tempo limite. Alguns pontos-chave a observar:

- CPU:
 - Uso da CPU da instância do EC2: verifique se a CPU não está saturada ou perto de 100%. A análise histórica pode ser feita via CloudWatch, 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 as [instâncias do EC2 com intermitência](#), certifique-se de que o saldo de crédito da CPU não tenha se esgotado. Essas informações estão disponíveis na CPUcreditBalance CloudWatch métrica.
 - Períodos curtos de alto uso da CPU 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 instâncias do Amazon EC2](#)
 - Em instâncias com o Driver de rede aprimorado ena, verifique as [estatísticas do ena](#) sobre tempo limite ou limites excedidos. As estatísticas a seguir são úteis para confirmar a saturação de limites de rede:
 - `bw_in_allowance_exceeded / bw_out_allowance_exceeded`: número de pacotes modelados devido ao throughput excessivo de entrada ou saída;
 - `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 de instâncias, NTP via VPC DNS. O limite é de 1024 pacotes por segundo para todos os serviços;
 - `pps_allowance_exceeded`: número de pacotes descartados devido à proporção excessiva de pacotes por segundo. O limite de PPS pode ser atingido quando o tráfego de rede consistir em milhares ou milhões de solicitações muito pequenas por segundo. 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: Tcpdump 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, tcpdump estava em execução e retornou:

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
    > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
    8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
    53962565, ack 177032945, win
    28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
    length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 1, win
    211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
    options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
    1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.], ack 6, win
    227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
    IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
    options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 8, win
    211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
    7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
```

```

1609428918.918307
  IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 9, win 211, options
  [nop,nop,TS val 27819441 ecr 3788576332], length 0
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel

```

A partir da saída acima, podemos confirmar que o handshake de três vias do TCP foi concluído em 222 microssegundos (918091 - 917869) e o comando ping foi enviado e retornado em 173 microssegundos (918295 - 918122).

Demorou 438 microssegundos (918307 - 917869) da solicitação ao fechamento da conexão. Esses resultados confirmam que os tempos de resposta da rede e do mecanismo são bons e que a investigação pode concentrar-se em outros componentes.

- No sistema operacional: `Strace` pode ajudar a identificar intervalos de tempo no nível do sistema operacional. A análise de aplicações reais seria muito mais extensa, e profilers ou depuradores especializados de aplicações são aconselhados. O exemplo a seguir mostra apenas se os componentes básicos do sistema operacional estão funcionando como esperado, caso contrário, investigações adicionais podem ser necessárias. Usando do mesmo comando PING do Redis 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"], 0x7fffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
  (+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
  (+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3
1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
  IPPROTO_IP) = 3
1609430221.717419

```


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 [compliance programs 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 no Amazon ElastiCache](#)
- [Privacidade do tráfego entre redes](#)
- [Gerenciamento de identidade e acesso no Amazon ElastiCache](#)
- [Validação de conformidade para o Amazon ElastiCache](#)
- [Resiliência no Amazon ElastiCache](#)
- [Segurança da infraestrutura no ElastiCache da AWS](#)
- [Atualizações de serviço em ElastiCache](#)
- [Vulnerabilidades e exposições comuns \(CVE\): vulnerabilidades de segurança abordadas no Redis ElastiCache](#)

Proteção de dados no Amazon ElastiCache

O [modelo de responsabilidade compartilhada](#) da AWS se aplica à proteção de dados no AWS ElastiCache (ElastiCache). Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos serviços da AWS que você usa. Para obter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#).

Para fins de proteção de dados, recomendamos que você proteja as credenciais da conta da AWS e configure as contas individuais com o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA [multi-factor authentication]) com cada conta.
- Use TLS para se comunicar com os recursos da AWS.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão nos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais armazenados no Amazon S3.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o ElastiCache ou outros serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Todos os dados inseridos no ElastiCache ou em outros serviços podem ser selecionados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

Tópicos

- [Segurança de dados no Amazon ElastiCache](#)

Segurança de dados no Amazon ElastiCache

Para ajudar a manter seus dados seguros, o Amazon ElastiCache e o Amazon EC2 fornecem mecanismos para proteger contra o acesso não autorizado aos seus dados no servidor.

O Amazon ElastiCache para Redis oferece recursos de criptografia para dados em caches que executam as versões do Redis 3.2.6 (programadas para EOL, consulte [Cronograma de fim de vida útil das versões do Redis](#)), 4.0.10 ou posterior:

- A criptografia em trânsito criptografa seus dados sempre que eles estão se movendo de um lugar para outro, como entre os nós no seu cluster ou entre seu cache e o aplicativo.
- A criptografia em repouso criptografa dados no disco durante as operações de sincronização e backup.

O Amazon ElastiCache para Redis dá suporte à autenticação de usuários com o IAM e o Redis AUTH, além da autorização das operações de usuários usando o controle de acesso por perfil (RBAC).

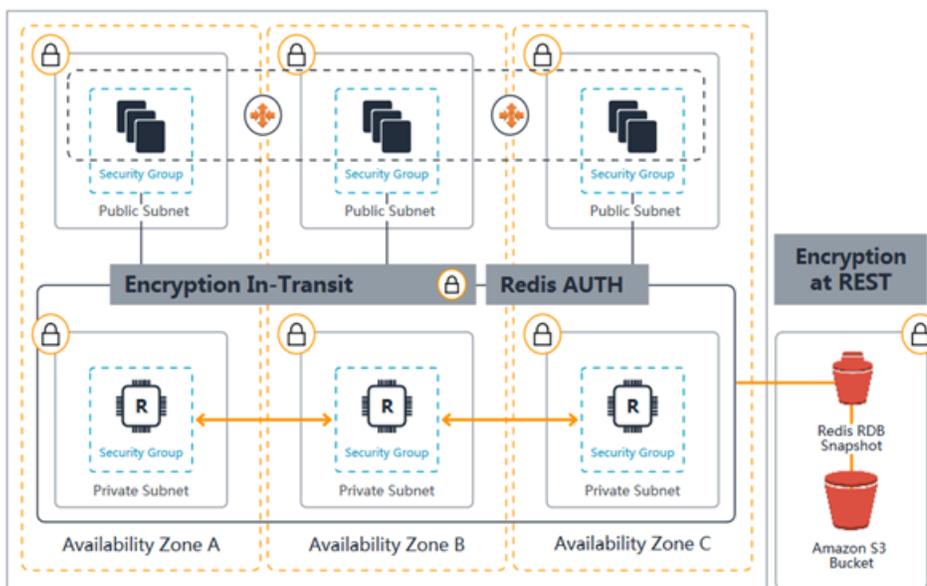


Diagrama de segurança do ElastiCache for Redis

Tópicos

- [Criptografia em trânsito \(TLS\) do ElastiCache](#)
- [Criptografia em repouso no ElastiCache](#)
- [Autenticação e autorização](#)

Criptografia em trânsito (TLS) do ElastiCache

Para ajudar a manter seus dados seguros, o Amazon ElastiCache e o Amazon EC2 fornecem mecanismos para proteger contra o acesso não autorizado aos seus dados no servidor. Ao fornecer a capacidade de criptografia em trânsito, o ElastiCache oferece uma ferramenta que pode ser usada para ajudar a proteger seus dados durante sua movimentação de um local para outro.

Todos os caches sem servidor têm criptografia em trânsito habilitada. Para clusters autoprojettados, você pode habilitar a criptografia em trânsito em um grupo de replicação configurando o parâmetro `TransitEncryptionEnabled` como `true` (CLI: `--transit-encryption-enabled`) ao criar o grupo de replicação. Isso pode ser feito ao criar o grupo de replicação usando o AWS Management Console, a AWS CLI ou a API do ElastiCache.

Tópicos

- [Visão geral da criptografia em trânsito](#)
- [Condições de criptografia em trânsito](#)
- [Práticas recomendadas de criptografia em trânsito](#)
- [Consulte também](#)
- [Habilitação da criptografia em trânsito](#)
- [Conectando-se à Amazon ElastiCache for Redis com criptografia em trânsito usando redis-cli](#)
- [Ativar criptografia em trânsito em um cluster Redis autoprojettado usando Python](#)
- [Práticas recomendadas ao habilitar a criptografia em trânsito](#)

Visão geral da criptografia em trânsito

A criptografia em trânsito do Amazon ElastiCache é um recurso que permite aumentar a segurança de seus dados em seus 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.

A criptografia em trânsito do ElastiCache implementa os seguintes recursos:

- Conexões de cliente criptografadas: as conexões do cliente com os nós de cache são criptografadas por TLS.
- Conexões de servidor criptografadas: os dados que se movem entre os nós em um cluster são criptografados.

- Autenticação do servidor: os clientes podem autenticar que estão conectados ao servidor certo.
- Autenticação do cliente: usando o recurso Redis AUTH, o servidor pode autenticar os clientes.

Condições de criptografia em trânsito

Lembre-se das seguintes restrições de criptografia em trânsito do Amazon ElastiCache quando você planejar sua implementação de cluster autoprojeto:

- A criptografia em trânsito é compatível em grupos de replicação que executam as versões 3.2.6, 4.0.10 e posteriores do Redis.
- A modificação da configuração de criptografia em trânsito, para um cluster existente, tem suporte em grupos de replicação que executam o Redis versão 7 e posterior.
- A criptografia em trânsito tem suporte somente em grupos de replicação em execução em uma Amazon VPC.
- A criptografia em trânsito tem suporte somente em grupos de replicação que estejam executando os tipos de nó a seguir.
 - R6g, R5, R4, R3
 - M6g, M5, M4, M3
 - T4g, T3, T2

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- A criptografia em trânsito é ativada configurando explicitamente o parâmetro `TransitEncryptionEnabled` como `true`.
- Verifique se o cliente de armazenamento em cache dá suporte à conectividade TLS e se você a habilitou na configuração do cliente.

Práticas recomendadas de criptografia em trânsito

- Como a criptografia e descriptografia dos dados requerem processamento nos endpoints, a implementação da criptografia em trânsito pode reduzir o desempenho. Compare seus dados com criptografia em trânsito e sem criptografia para determinar o impacto no desempenho da sua implementação.
- Como criar novas conexões pode ser caro, é possível reduzir o impacto na performance da criptografia em trânsito persistindo suas conexões TLS.

Consulte também

- [Criptografia em repouso no ElastiCache](#)
- [Autenticar com o comando Redis AUTH](#)
- [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#)
- [Segurança do Amazon VPCs e do ElastiCache](#)
- [Gerenciamento de identidade e acesso no Amazon ElastiCache](#)

Habilitação da criptografia em trânsito

Todos os caches sem servidor têm criptografia em trânsito habilitada. Em um cluster autoprotetido, é possível habilitar a criptografia em trânsito usando o AWS Management Console, a AWS CLI ou a API do ElastiCache.

Habilitação da criptografia em trânsito usando o AWS Management Console

Habilitação da criptografia em trânsito em um novo cluster autoprotetido 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 processo passo a passo, consulte o seguinte:

- [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)

Habilitação da criptografia em trânsito em um cluster autoprotetido 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

clientes Redis se conectem usando conexões criptografadas e não criptografadas. Depois de migrar todos os seus clientes Redis para usar conexões criptografadas, você pode modificar a configuração do cluster para definir o modo de criptografia em trânsito como `required`. Se o modo de criptografia em trânsito for definido como `required`, todas as conexões não criptografadas serão eliminadas e somente conexões criptografadas serão permitidas.

Etapa 1: defina seu modo de criptografia em trânsito como Preferred (Preferencial)

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Escolha Clusters do Redis nos Recursos do ElastiCache listados no painel de navegação, presente no lado esquerdo.
3. Escolha o Cluster do Redis que você deseja atualizar.
4. Escolha o menu suspenso Actions (Ações) e escolha Modify (Modificar).
5. Escolha Enable (Habilitar) em Encryption in transit (Criptografia em trânsito) na seção Security (Segurança).
6. Escolha Preferred (Preferencial) como Transit encryption mode (Modo de criptografia em trânsito).
7. Escolha Preview changes (Visualizar alterações) e salve suas alterações.

Depois de migrar todos os seus clientes Redis para usar conexões criptografadas:

Etapa 2: defina seu Transit encryption mode (Modo de criptografia em trânsito) como Required (Obrigatório)

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Escolha Clusters do Redis nos Recursos do ElastiCache listados no painel de navegação, presente no lado esquerdo.
3. Escolha o Cluster do Redis 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.

Habilitação da criptografia em trânsito usando o AWS CLI

Para habilitar a criptografia em trânsito ao criar um grupo de replicação do Redis usando a AWS CLI, use o parâmetro `transit-encryption-enabled`.

Habilitação da criptografia em trânsito em um cluster autoprojeto do Redis (modo cluster desabilitado) (CLI)

Use a operação da AWS CLI `create-replication-group` e os seguintes parâmetros para criar um grupo de replicação do Redis com réplicas com criptografia em trânsito:

Principais parâmetros:

- **--engine**: deve ser `redis`.
- **--engine-version**: deve 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 ver mais informações, consulte:

- [Criação de um grupo de replicação do Redis \(modo cluster desabilitado\) do princípio \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitação da criptografia em trânsito em um cluster autoprojeto do Redis (modo cluster habilitado) (CLI)

Use a operação da AWS CLI `create-replication-group` e os seguintes parâmetros para criar um grupo de replicação do Redis (modo cluster habilitado) com réplicas com criptografia em trânsito habilitada:

Principais parâmetros:

- **--engine**: deve ser `redis`.
- **--engine-version**: deve 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 ver mais informações, consulte:

- [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitação da criptografia em trânsito em um cluster existente usando o AWS CLI

Habilitar a criptografia em trânsito é um processo de duas etapas para o qual você deve primeiro definir o modo de criptografia de em trânsito como `preferred`. Esse modo permite que seus clientes Redis se conectem usando conexões criptografadas e não criptografadas. Depois de migrar todos os seus clientes Redis para usar conexões criptografadas, você pode modificar a configuração do cluster para definir o modo de criptografia em trânsito como `required`. Se o modo de criptografia em trânsito for definido como `required`, todas as conexões não criptografadas serão eliminadas e somente conexões criptografadas serão permitidas.

Use a operação da AWS CLI `modify-replication-group` e os seguintes parâmetros para atualizar um grupo de replicação do Redis (modo cluster habilitado) com réplicas com criptografia em trânsito desabilitada:

Como habilitar a criptografia em trânsito

1. Defina o modo de criptografia em trânsito como `preferred`, usando os seguintes parâmetros
 - **--transit-encryption-enabled**—Obrigatório.
 - **--transit-encryption-mode**: deve ser definido como `preferred`.
2. Defina o modo de criptografia em trânsito como `required`, usando os seguintes parâmetros:

- **--transit-encryption-enabled**—Obrigatório.
- **--transit-encryption-mode**: deve ser definido como `required`.

Conectando-se à Amazon ElastiCache for Redis com criptografia em trânsito usando `redis-cli`

Para acessar dados de ElastiCache quatro caches do Redis habilitados com criptografia em trânsito, você usa clientes que trabalham com Secure Socket Layer (SSL). Você também pode usar a `redis-cli` com TLS/SSL no Amazon Linux e no Amazon Linux 2. Se o cliente não der suporte ao TLS, você poderá usar o comando `stunnel` no host do cliente a fim de criar um túnel SSL para os nós do Redis.

Conexão criptografada com Linux

Para usar o `redis-cli` para se conectar a um cluster 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 `redis-cli`. Esse utilitário está incluído na distribuição do software Redis.
2. No prompt de comando da sua instância do EC2, digite os comandos apropriados para a versão do Linux que você está usando.

Amazon Linux 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.

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

Amazon Linux

Se estiver usando o Amazon Linux, digite o seguinte:

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

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 redis-cli, é recomendável executar o comando opcional `make-test`
4. Para se conectar a um cluster com criptografia e autenticação ativadas, digite este comando:

```
redis-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: `redis-cli`

```
redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
```

Conexão criptografada com stunnel

Para usar o redis-cli para se conectar a um cluster Redis habilitado com criptografia em trânsito usando stunnel, siga estas etapas.

1. Use o SSH para se conectar ao seu cliente e instalar o stunnel.

```
sudo yum install stunnel
```

2. Execute o comando a seguir para criar e editar arquivos '/etc/stunnel/redis-cli.conf' simultaneamente para adicionar um ElastiCache endpoint de cluster do Redis a um ou mais parâmetros de conexão, usando a saída fornecida abaixo como modelo.

```
vi /etc/stunnel/redis-cli.conf

fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[redis-cli]
  client = yes
  accept = 127.0.0.1:6379
  connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[redis-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, `redis-cli` e `redis-cli-replica`. Os parâmetros são definidos como a seguir:

- `client` (cliente) é definido com `sim` para especificar que essa instância de stunnel é um cliente.
- `accept` (aceitar) é definido com o IP do cliente. Neste exemplo, o primário é definido com o Redis padrão 127.0.0.1 na porta 6379. A réplica deve chamar uma porta diferente e ser definida como 6380. Você pode usar portas efêmeras 1024-65535. Para obter mais informações, consulte [Portas efêmeras](#) no Guia do usuário da Amazon VPC.

- `connect` (conectar) é definido como o endpoint do servidor Redis. Para ter mais informações, consulte [Localização de endpoints de conexão](#).

3. Inicie `stunnel`.

```
sudo stunnel /etc/stunnel/redis-cli.conf
```

Use o comando `netstat` para confirmar que os túneis começaram a funcionar.

```
sudo netstat -tulnp | grep -i stunnel
```

```
tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTEN
          3189/stunnel
tcp        0      0 127.0.0.1:6380          0.0.0.0:*               LISTEN
          3189/stunnel
```

4. Conecte-se ao nó Redis criptografado usando o endpoint local do túnel.

- Se nenhuma senha AUTH foi usada durante a ElastiCache criação do cluster Redis, este exemplo usa o `redis-cli` para se conectar ao servidor ElastiCache for Redis usando o caminho completo para `redis-cli`, no Amazon Linux:

```
/home/ec2-user/redis-stable/src/redis-cli -h localhost -p 6379
```

Se uma senha AUTH foi usada durante a criação do cluster do Redis, este exemplo usará a `redis-cli` para se conectar ao servidor Redis usando caminho completo para a `redis-cli`, no Amazon Linux:

```
/home/ec2-user/redis-stable/src/redis-cli -h localhost -p 6379 -a my-secret-password
```

OU

- Altere o diretório para `redis-stable` e faça o seguinte:

Se nenhuma senha AUTH foi usada durante a ElastiCache criação do cluster Redis, este exemplo usa o `redis-cli` para se conectar ao servidor ElastiCache for Redis usando o caminho completo para `redis-cli`, no Amazon Linux:

```
src/redis-cli -h localhost -p 6379
```

Se uma senha AUTH foi usada durante a criação do cluster do Redis, este exemplo usará a `redis-cli` para se conectar ao servidor Redis usando caminho completo para a `redis-cli`, no Amazon Linux:

```
src/redis-cli -h localhost -p 6379 -a my-secret-password
```

Este exemplo usa o Telnet para se conectar ao servidor Redis.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Para encerrar e fechar os túneis SSL, `kill` o processo de `stunnel`.

```
sudo kill stunnel
```

Ativar criptografia em trânsito em um cluster Redis autoprojetoado usando Python

O guia a seguir demonstrará como ativar a criptografia em trânsito em um cluster Redis 7.0 que foi originalmente criado com a criptografia em trânsito desativada. Os clientes TCP e TLS continuarão se comunicando com o cluster durante esse processo sem tempo de inatividade.

O Boto3 obterá as credenciais necessárias

(`aws_access_key_id`, `aws_secret_access_key` e `aws_session_token`) das variáveis de ambiente. Essas credenciais serão coladas com antecedência no mesmo terminal bash em que executaremos o `python3` para processar o código Python mostrado nesse guia. O código no exemplo abaixo foi um processo de uma instância do EC2 que foi lançada na mesma VPC que será usada para criar o ElastiCache Redis Cluster nela.

Note

- Os exemplos a seguir usam o SDK boto3 para operações de gerenciamento do ElastiCache (criação de cluster ou usuário) e redis-py/redis-py-cluster para manipulação de dados.
- Você deve usar pelo menos a versão boto3 (=~) 1.26.39 para usar a migração TLS on-line com a API de modificação de cluster.
- O ElastiCache oferece suporte à migração de TLS on-line somente para clusters Redis com versão 7.0 ou posterior. Portanto, se você tiver um cluster executando uma versão do Redis anterior à 7.0, precisará atualizar a versão Redis do seu cluster. Para obter mais informações sobre as diferenças entre as versões, consulte [Principais diferenças de comportamento e compatibilidade da versão](#).

Tópicos

- [Defina as constantes de string que iniciarão o ElastiCache Redis Cluster](#)
- [Defina as classes para a configuração do cluster](#)
- [Defina uma classe que representará o próprio cluster](#)
- [\(Opcional\) Crie uma classe wrapper para demonstrar a conexão do cliente com o cluster Redis](#)
- [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 Redis Cluster

Primeiro, vamos definir algumas constantes de string simples do Python que conterão os nomes das entidades da AWS necessárias para criar o cluster do ElastiCache, como `security-group`, `Cache Subnet group` e `default parameter group`. Todas essas entidades da AWS devem ser criadas com antecedência em sua conta da AWS 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 do Python que representarão a configuração de um cluster, que conterá metadados sobre o cluster, como a versão do 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 do Python que representarão o próprio cluster ElastiCache Redis. Essa classe terá um campo cliente que conterá um cliente boto3 para operações de gerenciamento do ElastiCache, como criar o cluster e consultar a API do ElastiCache.

```

import botocore.config
import boto3

# Create boto3 client
def init_client(region: str = "us-east-1"):
    config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
    init_request = dict()
    init_request["config"] = config
    init_request["service_name"] = "elasticache"
    init_request["region_name"] = region
    return boto3.client(**init_request)

class ElastiCacheClusterBase:
    def __init__(self, name: str):
        self.name = name
        self.elasticache_client = init_client()

    def get_first_replication_group(self):
        return self.elasticache_client.describe_replication_groups(
            ReplicationGroupId=self.name

```

```
    )["ReplicationGroups"][0]

def get_status(self) -> str:
    return self.get_first_replication_group()["Status"]

def get_transit_encryption_enabled(self) -> bool:
    return self.get_first_replication_group()["TransitEncryptionEnabled"]

def is_available(self) -> bool:
    return self.get_status() == "available"

def is_modifying(self) -> bool:
    return self.get_status() == "modifying"

def wait_for_available(self):
    while True:
        if self.is_available():
            break
        else:
            time.sleep(5)

def wait_for_modifying(self):
    while True:
        if self.is_modifying():
            break
        else:
            time.sleep(5)

def delete_cluster(self) -> bool:
    self.elasticache_client.delete_replication_group(
        ReplicationGroupId=self.name, RetainPrimaryCluster=False
    )

def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
    # generate api call to migrate the cluster to TLS preferred or to TLS required
    self.elasticache_client.modify_replication_group(
        ReplicationGroupId=self.name,
        TransitEncryptionMode=new_transit_encryption_mode,
        TransitEncryptionEnabled=True,
        ApplyImmediately=True,
    )
    self.wait_for_modifying()

class ElastiCacheClusterCME(ElastiCacheClusterBase):
```

```

def __init__(self, name: str):
    super().__init__(name)

    @classmethod
    def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
        config = config or ConfigCME()
        print(config)
        new_cluster = ElastiCacheClusterCME(config.name)
        launch_request = config.create_launch_request()
        new_cluster.elasticache_client.create_replication_group(**launch_request)
        new_cluster.wait_for_available()
        return new_cluster

    def get_configuration_endpoint(self) -> str:
        return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]

#Since the code can throw exceptions, we define this class to make the code more
readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
    def __init__(self, config: ConfigCME = None):
        self.config = config or ConfigCME()

    def __enter__(self) -> ElastiCacheClusterCME:
        self.cluster = ElastiCacheClusterCME.launch(self.config)
        return self.cluster

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.cluster.delete_cluster()

```

(Opcional) Crie uma classe wrapper para demonstrar a conexão do cliente com o cluster Redis

Agora, vamos criar uma classe de wrapper para o cliente `redis-py-cluster`. Essa classe de wrapper será compatível com o pré-preenchimento do cluster com algumas chaves e, em seguida, com a execução de comandos `get` aleatórios repetidos.

Note

Essa é uma etapa opcional, mas simplifica o código da função principal que vem em uma etapa posterior.

```
import redis
import random
from time import perf_counter_ns, time

class DowntimeTestClient:
    def __init__(self, client):
        self.client = client

        # num of keys prefilled
        self.prefilled = 0
        # percent of get above prefilled
        self.percent_get_above_prefilled = 10 # nil result expected when get hit above
prefilled
        # total downtime in nano seconds
        self.downtime_ns = 0
        # num of success and fail operations
        self.success_ops = 0
        self.fail_ops = 0
        self.connection_errors = 0
        self.timeout_errors = 0

    def replace_client(self, client):
        self.client = client

    def prefill_data(self, timelimit_sec=60):
        end_time = time() + timelimit_sec
        while time() < end_time:
            self.client.set(self.prefilled, self.prefilled)
            self.prefilled += 1

    # unsuccessful operations throw exceptions
    def _exec(self, func):
        try:
            start_ns = perf_counter_ns()
            func()
            self.success_ops += 1
            elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
            # upon succesful execution of func
            # reset random_key to None so that the next command
            # will use a new random key
            self.random_key = None
```

```
except Exception as e:
    elapsed_ns = perf_counter_ns() - start_ns
    self.downtime_ns += elapsed_ns
    # in case of failure- increment the relevant counters so that we will keep
track
    # of how many connection issues we had while trying to communicate with
    # the cluster.
    self.fail_ops += 1
    if e.__class__ is redis.exceptions.ConnectionError:
        self.connection_errors += 1
    if e.__class__ is redis.exceptions.TimeoutError:
        self.timeout_errors += 1

def _repeat_exec(self, func, seconds):
    end_time = time() + seconds
    while time() < end_time:
        self._exec(func)

def _new_random_key_if_needed(self, percent_above_prefilled):
    if self.random_key is None:
        max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
        return random.randint(0, max)
    return self.random_key

def _random_get(self):
    key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
    result = self.client.get(key)
    # we know the key was set for sure only in the case key < self.prefilled
    if key < self.prefilled:
        assert result.decode("UTF-8") == str(key)

def repeat_get(self, seconds=60):
    self._repeat_exec(self._random_get, seconds)

def get_downtime_ms(self) -> int:
    return self.downtime_ns // 10 ** 6

def do_get_until(self, cond_check):
    while not cond_check():
        self.repeat_get()
    # do one more get cycle once condition is met
```

```
self.repeat_get()
```

Crie a função principal que demonstra o processo de alteração da configuração de criptografia em trânsito

Agora, vamos definir a função principal, que fará o seguinte:

1. Crie o cluster usando o cliente boto3 ElastiCache.
2. Inicialize o cliente `redis-py-cluster` que se conectará ao cluster com uma conexão TCP clara sem TLS.
3. O cliente `redis-py-cluster` preenche o cluster com alguns dados.
4. O cliente boto3 acionará a migração de TLS de não TLS para TLS preferencial.
5. Enquanto o cluster estiver sendo migrado para o TLS Preferred, o cliente TCP `redis-py-cluster` enviará operações `get` repetidas para o cluster até que a migração seja concluída.
6. Após a conclusão da migração para o TLS Preferred, afirmaremos que o cluster é compatível com a criptografia em trânsito. Depois, criaremos um cliente `redis-py-cluster` que se conectará ao cluster com TLS.
7. Enviaremos alguns comandos `get` usando o novo cliente TLS e o antigo cliente TCP.
8. O cliente boto3 acionará a migração de TLS de TLS Preferred para TLS obrigatório.
9. Enquanto o cluster estiver sendo migrado para o TLS obrigatório, o cliente TLS `redis-py-cluster` enviará operações `get` repetidas para o cluster até que a migração seja concluída.

```
import redis

def init_cluster_client(
    cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
DowntimeTestClient:
    # we must use for the host name the cluster configuration endpoint.
    redis_client = redis.RedisCluster(
        host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
        socket_connect_timeout=0.1
    )
    test_client = DowntimeTestClient(redis_client)
    if prefill_data:
        test_client.prefill_data()
    return test_client

if __name__ == '__main__':
```

```
config = ConfigCME(TLS=False, instance_type="cache.m5.large")

with ElastiCacheCMEManager(config) as cluster:
    # create a client that will connect to the cluster with clear tcp connection
    test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

    # migrate the cluster to TLS Preferred
    cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

    # do repeated get commands until the cluster finishes the migration to TLS
    Preferred
    test_client_tcp.do_get_until(cluster.is_available)

    # verify that in transit encryption is enabled so that clients will be able to
    connect to the cluster with TLS
    assert cluster.get_transit_encryption_enabled() == True

    # create a client that will connect to the cluster with TLS connection.
    # we must first make sure that the cluster indeed supports TLS
    test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

    # by doing get commands with the tcp client for 60 more seconds
    # we can verify that the existing tcp connection to the cluster still works
    test_client_tcp.repeat_get(seconds=60)

    # do get commands with the new TLS client for 60 more seconds
    test_client_tcp.repeat_get(seconds=60)

    # migrate the cluster to TLS required
    cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

    # from this point the tcp clients will be disconnected and we must not use them
    anymore.
    # do get commands with the TLS client until the cluster finishes migration to
    TLS required mode.
    test_client_tls.do_get_until(cluster.is_available)
```

Práticas recomendadas ao habilitar a criptografia em trânsito

Antes de ativar a criptografia em trânsito: verifique se você tem o tratamento adequado dos registros DNS

Note

Estamos alterando e excluindo endpoints antigos durante esse processo. O uso incorreto dos endpoints pode fazer com que o cliente 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 DNS por nó antigos são mantidos e os novos registros DNS por nó são gerados em um formato diferente. Os clusters habilitados para TLS usam um formato de registros DNS diferente dos clusters habilitados para não TLS. O ElastiCache manterá os dois registros DNS quando um cluster for configurado no modo de criptografia preferencial para que aplicativos e outros clientes Redis possam alternar entre eles. As seguintes alterações nos registros DNS ocorrem durante o processo de migração do TLS:

Descrição das alterações nos registros DNS que ocorrem ao ativar a criptografia em trânsito

Para clusters CME

Quando um cluster é definido como “modo de criptografia em trânsito: preferencial”:

- Os endpoints do cluster original para o cluster habilitados para não TLS permanecerão ativos. Não haverá tempo de inatividade quando o cluster for reconfigurado do modo de criptografia TLS 'nenhum' para 'preferencial'.
- Novos endpoints TLS Redis serão gerados quando o cluster for definido no modo de TLS preferencial. Esses novos endpoints terão os mesmos IPs dos antigos (não TLS).
- O novo endpoint de configuração do TLS Redis será exposto no console do ElastiCache e na resposta à API `describe-replication-group`.

Quando um cluster é definido como “modo de criptografia em trânsito: obrigatório”:

- Os endpoints habilitados para não TLS antigos serão excluídos. Não haverá tempo de inatividade dos endpoints do cluster TLS.
- Você pode recuperar um novo `cluster-configuration-endpoint` do console do ElastiCache ou da API `describe-replication-group`.

Para clusters CMD com Failover automático ativado ou Failover automático desativado

Quando um grupo de replicação é definido como “modo de criptografia em trânsito: preferencial”:

- O endpoint primário original e o endpoint do leitor para um cluster habilitado para não TLS permanecerão ativos.
- Os novos endpoints TLS primários e do leitor serão gerados quando o cluster for definido no modo de TLS Preferred. Esses novos endpoints terão os mesmos IPs dos antigos (não TLS).
- Os novos endpoints primário e de leitor serão expostos no console do ElastiCache e na resposta à API `describe-replication-group`.

Quando o grupo de replicação é definido como “modo de criptografia em trânsito: obrigatório”:

- O endpoint primário original e o endpoint do leitor para um cluster habilitado para não TLS permanecerão ativos.
- Os endpoints primários e de leitura não TLS antigos serão excluídos. Não haverá tempo de inatividade dos endpoints do cluster TLS.
- Você pode recuperar os novos endpoints primário e do leitor no console do ElastiCache ou na API `describe-replication-group`.

Uso sugerido dos registros DNS

Para clusters CME

- Use o endpoint de configuração do cluster em vez dos registros DNS por nó no código do seu aplicativo. Não é recomendável usar nomes DNS por nó diretamente, pois eles podem mudar quando fragmentos forem adicionados ou removidos.
- 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 a API `describe-replication-group` [conforme demonstrado acima (em negrito)] e use o DNS que você obtiver em resposta a partir desse momento.

Para clusters CMD com Failover Automático ativado

- Use o endpoint primário e o endpoint do leitor em vez dos nomes DNS por nó no código do seu aplicativo, pois os nomes DNS por nó antigos são excluídos e os novos são gerados ao migrar o cluster do não TLS para o TLS preferencial. Não é recomendável usar nomes DNS por nó diretamente, pois você pode adicionar réplicas ao seu cluster no futuro. Além disso, quando o Failover Automático está ativado, as funções do cluster primário e das réplicas são alteradas automaticamente pelo serviço ElastiCache. Sugere-se usar o endpoint primário e o endpoint do leitor para ajudar você a acompanhar essas alterações. Por fim, usar o endpoint do leitor ajudará você a distribuir as leituras das réplicas igualmente entre as réplicas no cluster.
- Ter o endpoint primário e o endpoint do leitor codificados em seu aplicativo é uma prática ruim, pois isso pode ser alterado durante o processo de migração do TLS. Depois que a alteração da migração para o TLS preferencial for concluída, consulte o endpoint primário e o endpoint do leitor com a API describe-replication-group e use o DNS que você obtiver em resposta a partir desse momento. Dessa forma, você poderá acompanhar as mudanças nos endpoints de forma dinâmica.

Para cluster CMD com Failover automático ativado

- Use o endpoint primário e o endpoint do leitor em vez dos nomes DNS por nó no código do seu aplicativo. Quando o Failover automático está desativado, o ajuste de escala, a aplicação de patches, o failover e outros procedimentos que são gerenciados automaticamente pelo serviço ElastiCache quando o Failover automático está ativado são feitos por você. Isso facilita seu acompanhamento manual dos diferentes endpoints. Como os nomes DNS por nó antigos são excluídos e os novos são gerados ao migrar o cluster de não TLS para TLS preferencial, não use os nomes DNS por nó diretamente. Isso é obrigatório para que os clientes possam se conectar ao cluster durante a migração para TLS. Além disso, você se beneficiará de distribuir uniformemente as leituras entre as réplicas ao usar o endpoint do leitor e acompanhar os registros de DNS ao adicionar ou excluir réplicas do cluster.
- Ter o endpoint de configuração do cluster codificado em seu aplicativo é uma prática ruim, pois ele pode ser alterado durante o processo de migração TLS.

Durante a criptografia em trânsito: preste atenção quando o processo de migração terminar

A alteração do modo de criptografia em trânsito não é imediata e pode levar algum tempo. Isso é especialmente verdade para clusters grandes. Somente quando o cluster conclui a migração para o TLS preferencial é que ele pode aceitar e servir conexões TCP e TLS. Portanto, você não deve criar clientes que tentarão estabelecer conexões TLS com o cluster até que a criptografia em trânsito seja concluída.

Há várias maneiras de ser notificado quando a criptografia em trânsito é concluída com êxito ou falha: (não mostrado no exemplo de código acima):

- Usar o serviço SNS para receber uma notificação quando a criptografia for concluída
- Usar a API `describe-events` que emitirá um evento quando a criptografia for concluída
- Mensagem no console do ElastiCache informando que a criptografia foi concluída

Você também pode implementar a lógica em seu aplicativo para saber se a criptografia foi concluída. No exemplo acima, vimos várias maneiras de garantir que o cluster conclua a migração:

- Esperar até o início do processo de migração (o status do cluster muda para “modificando”) e esperar até que a modificação seja concluída (o status do cluster volta para “disponível”)
- Verificar se `transit_encryption_enabled` do cluster está definido como Verdadeiro consultando a API `describe-replication-group`.

Depois de ativar a criptografia em trânsito: verifique se os clientes que você usa estão configurados corretamente

Enquanto o cluster estiver no modo de TLS preferencial, seu aplicativo deve abrir conexões TLS com o cluster e usar somente essas conexões. Dessa forma, seu aplicativo não sofrerá tempo de inatividade ao ativar a criptografia em trânsito. Você pode garantir que não haja conexões TCP mais claras com o mecanismo Redis usando o comando Redis `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
```

Criptografia em repouso no ElastiCache

Para ajudar a manter seus dados seguros, o Amazon ElastiCache e o Amazon S3 fornecem maneiras diferentes de restringir o acesso aos dados no cache. Para obter mais informações, consulte [Segurança do Amazon VPCs e do ElastiCache](#) e [Gerenciamento de identidade e acesso no Amazon ElastiCache](#).

A criptografia em repouso do ElastiCache é um recurso para aumentar a segurança dos dados criptografando os 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 unidades de estado sólido (SSDs) em clusters habilitados para divisão de dados em camada são sempre criptografados por padrão.

O ElastiCache oferece criptografia padrão (gerenciada pelo serviço) em repouso, bem como a capacidade de usar as próprias chaves do AWS KMS simétricas gerenciadas pelo cliente no [AWS Key Management Service \(KMS\)](#). Quando o backup do cache for feito, em opções de criptografia, escolha se você deseja usar a chave de criptografia padrão ou uma chave gerenciada pelo cliente. Para obter mais informações, consulte [Ativar criptografia em repouso](#).

Note

A criptografia padrão (gerenciada por serviço) é a única opção disponível nas regiões GovCloud (EUA).

Important

A habilitação da criptografia em repouso em um cluster Redis autoprojeto existente envolve a exclusão do grupo de replicação existente após a execução do backup e a restauração no grupo de replicação.

A criptografia em repouso só pode ser habilitada em um cache quando é criada. Como a criptografia e a descriptografia dos dados requerem processamento, a ativação da criptografia em repouso pode

afetar o desempenho durante essas operações. Compare seus dados com e sem criptografia em repouso para determinar o impacto no desempenho para seus casos de uso.

Tópicos

- [Condições da criptografia em repouso](#)
- [Uso de chaves gerenciadas pelo cliente do KMS da AWS](#)
- [Ativar criptografia em repouso](#)
- [Consulte também:](#)

Condições da criptografia em repouso

Lembre-se das seguintes restrições de criptografia em repouso do ElastiCache quando você planejar sua implementação de criptografia em repouso do ElastiCache:

- A criptografia em repouso é compatível em grupos de replicação que estejam executando as versões do Redis (3.2.6 programadas para EOL, consulte [Cronograma de fim de vida útil das versões do Redis](#)) 4.0.10 ou posteriores.
- A criptografia em repouso tem suporte somente em grupos de replicação em execução em uma Amazon VPC.
- A criptografia em repouso é compatível somente com grupos de replicação que estejam executando os tipos de nó a seguir.
 - 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 em SSD serão criptografados independentemente de a criptografia em repouso estar ativada ou não.

- A opção de usar chave gerenciada pelo cliente para criptografia em repouso não está disponível nas regiões da AWS GovCloud (us-gov-east-1 e us-gov-west-1).
- Se um cluster estiver usando um tipo de nó da família r6gd, os dados armazenados em SSD serão criptografados com a chave do AWS KMS gerenciada pelo cliente escolhida (ou criptografia gerenciada por serviço em regiões AWS GovCloud).

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.

Uso de chaves gerenciadas pelo cliente do KMS da AWS

O ElastiCache dá suporte a chaves do AWS KMS simétricas gerenciadas pelo cliente para criptografia em repouso. Chaves KMS gerenciadas pelo cliente são chaves de criptografia que você cria, detém e gerencia na sua conta da AWS. Para obter mais informações, consulte [Chaves KMS da AWS](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS. As chaves devem ser criadas no AWS KMS para serem usadas com o ElastiCache.

Para saber como criar chaves de raiz do KMS da AWS, consulte [Criação de chaves](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS.

O ElastiCache permite a integração com o AWS KMS. Para obter mais informações, consulte [Uso de concessões](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS. Nenhuma ação do cliente é necessária para ativar a integração do Amazon ElastiCache com o KMS da AWS.

A chave de condição `kms:ViaService` limita o uso de uma chave do KMS da AWS (chave do KMS) para solicitações provenientes de serviços da AWS específicos. Para usar `kms:ViaService` com o ElastiCache, inclua ambos os nomes do `ViaService` no valor da chave de condição: `elasticache.AWS_region.amazonaws.com` e `dax.AWS_region.amazonaws.com`. Para obter mais informações, consulte [kms:ViaService](#).

Você pode usar o [CloudTrail da AWS](#) para rastrear as solicitações que o Amazon ElastiCache envia para o AWS Key Management Service em seu nome. Todas as chamadas de API para o AWS Key Management Service relacionadas a chaves gerenciadas pelo cliente têm logs do CloudTrail correspondentes. Também é possível ver as concessões criadas pelo ElastiCache fazendo a chamada de API do KMS [ListGrants](#).

Assim que um grupo de replicação é criptografado usando a chave gerenciada pelo cliente, todos os backups do grupo de replicação são criptografados da seguinte maneira:

- Os backups diários automáticos são criptografados usando a chave gerenciada pelo cliente associada ao cluster.
- O backup final criado quando o grupo de replicação é excluído também é criptografado usando a chave gerenciada pelo cliente associada ao grupo de replicação.
- Os backups criados manualmente são criptografados por padrão para usar a chave do KMS associada ao grupo de replicação. Você pode substituir escolhendo outra chave gerenciada pelo cliente.
- Por padrão, a cópia de um backup equivale a usar uma chave gerenciada pelo cliente associada ao backup de origem. Você pode substituir escolhendo outra chave gerenciada pelo cliente.

Note

- As chaves gerenciadas pelo cliente não podem ser usadas ao exportar backups para o bucket do Amazon S3 selecionado. No entanto, todos os backups exportados para o Amazon S3 são criptografados usando [Criptografia do lado do servidor](#). Você pode optar por copiar o arquivo de backup para um novo objeto do S3 e criptografar usando uma chave do KMS gerenciada pelo cliente, copiar o arquivo para outro bucket do S3 configurado com a criptografia padrão usando uma chave do KMS ou alterar uma opção de criptografia no próprio arquivo.
- Você também pode usar chaves gerenciadas pelo cliente para criptografar backups criados manualmente para grupos de replicação que não usem chaves gerenciadas pelo cliente para criptografia. Com essa opção, o arquivo de backup armazenado no Amazon S3 é criptografado usando uma chave do KMS, embora os dados não sejam criptografados no grupo de replicação original.

A restauração de um backup permite escolher entre as opções de criptografia disponíveis, semelhantes às opções de criptografia disponíveis ao criar um novo grupo de replicação.

- Se você excluir a chave ou [desabilitá-la](#) e [revogar as concessões](#) da chave que usou para criptografar um cache, o cache ficará irrecoverável. Em outras palavras, ele não poderá ser modificado nem recuperado depois de uma falha de hardware. AWS O KMS exclui as chaves raiz somente depois de 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 mudança de chaves automática preserva as propriedades de suas chaves raiz do KMS da AWS, portanto, a rotação não afeta sua capacidade de acessar os dados do ElastiCache. Os caches do Amazon ElastiCache não dão suporte à mudança de chaves manual, que envolve a criação de uma nova chave raiz e a atualização de eventuais referências à chave anterior. Para saber mais, consulte [Alternância de chaves do KMS da AWS](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS.
- A criptografia de um cache do ElastiCache usando a chave do KMS requer uma concessão por cache. Essa concessão é usada durante toda a vida útil do cache. Além disso, uma concessão por backup é usada durante a criação do backup. Essa concessão é retirada assim que o backup é criado.
- Para obter mais informações sobre concessões e limites do KMS da AWS, consulte [Limites](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS.

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 habilitar a criptografia em repouso ao criar um cache do ElastiCache. Você pode fazer isso usando o AWS Management Console, a AWS CLI ou a API do ElastiCache.

Ao criar um cache, você pode escolher uma das seguintes opções:

- Default (Padrão) – Esta opção usa a criptografia gerenciada pelo serviço em repouso.
- Customer managed key (Chave gerenciada pelo cliente): esta opção permite fornecer o ID/ARN do KMS da AWS para criptografia em repouso.

Para saber como criar chaves de raiz do KMS da AWS, consulte [Criar chaves](#) no Guia do desenvolvedor do serviço de gerenciamento de chaves da AWS.

Sumário

- [Habilitação da criptografia em repouso usando o AWS Management Console](#)
- [Habilitação da criptografia em repouso usando o AWS CLI](#)

Habilitação da criptografia em repouso em um cluster Redis autoprojeto existente

Só é possível ativar a criptografia em repouso ao criar um grupo de replicação do Redis. 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 [Excluir um cluster](#) ou [Exclusão de um grupo de replicação](#).

Habilitação da 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 chave do KMS de propriedade da AWS é usada para criptografar dados. Para escolher a própria chave do AWS KMS, faça as seguintes seleções:

- Expanda a seção Visualizar configurações padrão.
- Escolha Personalizar configurações padrão na seção Visualizar configurações padrão.
- Escolha Personalize suas configurações de segurança na seção Segurança.
- Escolha CMK gerenciada pelo cliente na configuração Chave de criptografia.
- Selecione uma chave na configuração Chave AWS KMS.

Habilitação da criptografia em repouso em um cluster 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 CMK gerenciada pelo cliente.

Para o procedimento passo a passo, consulte o seguinte:

- [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)

Habilitação da criptografia em repouso usando o AWS CLI

Para ativar a criptografia em repouso ao criar um cluster Redis usando a AWS CLI, use o parâmetro `--at-rest-encryption-enabled` ao criar um grupo de replicação.

Habilitação da criptografia em repouso em um cluster do Redis (modo cluster desabilitado) (CLI)

A operação a seguir cria o grupo de replicação do Redis (modo cluster desabilitado) `my-classic-rg` com três nós (`--num-cache-clusters`): um primário e duas réplicas de leitura. A criptografia em repouso é ativada 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 `redis`.
- `--engine-version`: deve ser 3.2.6, 4.0.10 ou posterior.
- `--at-rest-encryption-enabled`: obrigatório para habilitar a criptografia em repouso.

Example 1: cluster do Redis (modo cluster desabilitado) 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:

- [Criação de um grupo de replicação do Redis \(modo cluster desabilitado\) do princípio \(AWS CLI\)](#)
- [create-replication-group](#)

Habilitação da criptografia em repouso em um cluster do Redis (modo cluster habilitado) (CLI)

A operação a seguir cria o grupo de replicação do Redis (modo cluster habilitado) `my-clustered-rg` com três grupos de nó ou fragmentos (`--num-node-groups`). Cada um deles tem três nós: um primário e duas réplicas de leitura (`--replicas-per-node-group`). A criptografia em repouso é ativada 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 `redis`.
- **--engine-version**: deve 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.

Example 2: um cluster do Redis (modo cluster habilitado)

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-clustered-rg \  
  --replication-group-description "redis clustered cluster" \  
  --cache-node-type cache.m3.large \  
  --num-node-groups 3 \  
  --replicas-per-node-group 2 \  
  --engine redis \  
  --engine-version 6.2 \  
  --at-rest-encryption-enabled \  
  --cache-parameter-group default.redis6.x.cluster.on
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id my-clustered-rg ^  
  --replication-group-description "redis clustered cluster" ^  
  --cache-node-type cache.m3.large ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --engine redis ^  
  --engine-version 6.2 ^  
  --at-rest-encryption-enabled ^  
  --cache-parameter-group default.redis6.x.cluster.on
```

Para obter informações adicionais, consulte:

- [Criação de um grupo de replicação no Redis \(modo cluster habilitado\) do princípio \(AWS CLI\)](#)
- [create-replication-group](#)

Consulte também:

- [Segurança do Amazon VPCs e do ElastiCache](#)
- [Gerenciamento de identidade e acesso no Amazon ElastiCache](#)

Autenticação e autorização

O ElastiCache dá suporte à autenticação de usuários usando o IAM e o comando Redis AUTH, além da autorização das operações de usuários usando o controle de acesso por perfil (RBAC).

Tópicos

- [Regras de controle de acesso com base em função \(RBAC\)](#)
- [Autenticar com o comando Redis AUTH](#)
- [Desabilitar o controle de acesso em um cache do ElastiCache para Redis](#)

Regras de controle de acesso com base em função (RBAC)

Em vez de autenticar usuários com o comando Redis AUTH conforme descrito em [Autenticar com o comando Redis AUTH](#), no Redis 6.0 em diante, você pode usar um recurso chamado Role-Based Access Control (RBAC – Controle de acesso baseado em função). O RBAC também é a única maneira de controlar o acesso a caches sem servidor.

Diferentemente do Redis AUTH, no qual todos os clientes autenticados têm acesso total em caso de autenticação do token, o RBAC permite controlar o acesso ao cache por meio dos grupos de usuários. Esses grupos de usuários foram projetados como uma maneira de organizar o acesso a caches.

Com o RBAC, você cria usuários e atribui a eles permissões específicas usando uma cadeia de acesso, conforme descrito a seguir. Você atribui os usuários a grupos de usuários alinhados a uma função específica (administradores, recursos humanos) implantados em um ou mais chats do ElastiCache para Redis. Fazendo isso, você pode estabelecer limites de segurança entre clientes usando o mesmo cache ou caches do Redis e impedir que os clientes acessem os dados uns dos outros.

O RBAC foi projetado para oferecer suporte à introdução da [ACL do Redis](#) no Redis 6. Quando você usa o RBAC com o cache do ElastiCache para Redis, existem algumas limitações:

- Não é possível especificar senhas em uma string de acesso. Você define as senhas com chamadas a [CreateUser](#) ou [ModifyUser](#).
- 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, ou seja, o nome do comando antes de ele ser renomeado.

- Não é possível usar o comando `reset` como parte de uma string de acesso. Você especifica senhas com parâmetros de API e o ElastiCache para Redis gerencia senhas. Assim, você não pode usar `reset` porque ele removeria todas as senhas de um usuário.
- O Redis 6 apresenta o comando [ACL LIST](#). Esse comando retorna uma lista de usuários junto com as regras da ACL aplicadas a cada usuário. O ElastiCache para Redis oferece suporte ao comando `ACL LIST`, mas não inclui suporte para hashes de senha como o Redis faz. Com o ElastiCache para Redis, 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 de leitura suportados pelo ElastiCache para Redis incluem [ACL WHOAMI](#), [ACL USERS](#) e [ACL CAT](#). O ElastiCache para Redis não oferece suporte a nenhum outro comando ACL baseado em gravação.

- As limitações a seguir aplicam-se:

Recurso	Máximo permitido
Usuários por grupo de usuários	100
Número de DPUs	1000
Número de grupos de usuários	100

O uso do RBAC com o ElastiCache para Redis é descrito com mais detalhes a seguir.

Tópicos

- [Especificação de permissões usando uma string de acesso](#)
- [Aplicação do RBAC a um cache para ElastiCache para Redis](#)
- [Migração do Redis AUTH para o RBAC](#)
- [Migração do RBAC para o Redis AUTH](#)
- [Alternância automática de senhas para os usuários](#)
- [Autenticação com o IAM](#)

Especificação de permissões usando uma string de acesso

Para especificar permissões para um cache do ElastiCache para Redis, você cria uma string de acesso e a atribui a um usuário, usando a AWS CLI ou o AWS Management Console.

As strings de acesso são definidas como uma lista de regras delimitadas por espaço que são aplicadas ao usuário. Elas definem quais comandos um usuário pode executar e em quais chaves um usuário pode operar. Para executar um comando, um usuário deve ter acesso ao comando que está sendo executado e todas as chaves que estão sendo acessadas pelo comando. Regras são aplicadas da esquerda para a direita cumulativamente, e uma string mais simples pode ser usada em vez da fornecida se houver redundâncias na string fornecida.

Para obter informações sobre a sintaxe das regras ACL, consulte [ACL](#).

No exemplo a seguir, a string de acesso representa um usuário ativo com acesso a todas as chaves e comandos disponíveis.

```
on ~* +@all
```

A sintaxe da cadeia de acesso é dividida da seguinte forma:

- `on`: o usuário é um usuário ativo.
- `~*`: o acesso é dado a todas as chaves disponíveis.
- `+@all`: o acesso é dado a todos os comandos disponíveis.

As configurações anteriores são as menos restritivas. Você pode modificar essas configurações para torná-las mais seguras.

No exemplo a seguir, a string de acesso representa um usuário com acesso restrito ao acesso de leitura em chaves que começam com o keypace "app::"

```
on ~app::* -@all +@read
```

Você pode refinar mais essas permissões listando comandos aos quais o usuário tem acesso:

`+command1`: o acesso do usuário aos comandos é limitado a *command1*.

`+@category`: o acesso do usuário é limitado a uma categoria de comandos.

Para obter informações sobre como atribuir uma string de acesso a um usuário, consulte [Criação de usuários e grupos de usuários com o console e a CLI](#).

Se você estiver migrando uma workload existente para o ElastiCache, poderá recuperar a string de acesso chamando `ACL LIST`, excluindo o usuário e quaisquer hashes de senha.

Para o Redis versão 6.2 e posterior, a seguinte sintaxe de cadeia de caracteres de acesso também tem suporte:

- `&*`: o acesso é dado a todos os canais disponíveis.

Para o Redis versão 7.0 e posterior, a seguinte sintaxe de cadeia de caracteres de acesso também tem suporte:

- `|`: pode ser usado para bloquear subcomandos (por exemplo, `"-config|set"`).
- `%R~<pattern>`: adicione o padrão de chave de leitura especificado. Comporta-se de forma semelhante ao padrão de chave normal, mas só concede permissão para ler chaves que correspondam ao padrão fornecido. Para obter mais informações, consulte [permissões de chave](#).
- `%W~<pattern>`: adicione o padrão de chave de gravação especificado. Comporta-se de forma semelhante ao padrão de chave normal, mas só concede permissão para gravar em chaves que correspondam ao padrão fornecido. Para obter mais informações, consulte [permissões de chave](#).
- `%RW~<pattern>`: alias para `~<pattern>`.
- `(<rule list>)`: crie um novo seletor para combinar as regras. Os seletores são avaliados após as permissões do usuário e são avaliados de acordo com a ordem em que são definidos. Se um comando corresponder às permissões do usuário ou a qualquer seletor, ele será permitido. Para obter mais informações, consulte [Seletores de ACL](#).
- `clearselectors`: exclua todos os seletores associados ao usuário.

Aplicação do RBAC a um cache para ElastiCache para Redis

Para usar o ElastiCache para Redis RBAC, siga estas etapas:

1. Crie um ou mais usuários.
2. Crie um grupo de usuários e adicione usuários ao grupo.
3. Atribua o grupo de usuários a um cache que tenha criptografia em trânsito habilitada.

Essas etapas estão descritas em detalhes a seguir.

Tópicos

- [Criação de usuários e grupos de usuários com o console e a CLI](#)
- [Gerenciamento de grupos de usuários com o console e a CLI](#)
- [Atribuição de grupos de usuários a caches sem servidor](#)
- [Atribuição de grupos de usuários a grupos de replicação](#)

Criação de usuários e grupos de usuários com o console e a CLI

As informações de usuário para usuários do RBAC são um ID de usuário, nome de usuário e, opcionalmente, uma senha e uma string de acesso. A string de acesso fornece o nível de permissão em chaves e comandos. O ID de usuário é exclusivo para o usuário e o nome de usuário é o que é passado para o mecanismo.

Certifique-se de que as permissões do usuário fornecidas fazem sentido com a finalidade pretendida do grupo de utilizadores. Por exemplo, se você criar um grupo de usuários chamado `Administrators`, qualquer usuário que você adicionar a esse grupo deve ter sua string de acesso definida para acesso total a chaves e comandos. Para usuários em um grupo de usuários e-commerce, você pode definir suas cadeias de acesso para acesso somente leitura.

O ElastiCache configura automaticamente um usuário padrão com ID de usuário e nome de usuário `default` e o adiciona a todos os grupos de usuários. Você não pode excluir ou modificar esse usuário. Esse usuário destina-se à compatibilidade com o comportamento padrão das versões anteriores do Redis e tem uma string de acesso que permite chamar todos os comandos e acessar todas as chaves.

Para adicionar um controle de acesso indicado a um cache, substitua esse usuário padrão por um novo que não esteja habilitado ou use uma senha forte. Para alterar o usuário padrão, crie um novo usuário com o nome de usuário definido como `default`. Em seguida, você pode trocá-lo pelo usuário padrão original.

Os seguintes procedimentos mostram como trocar o usuário `default` original por outro usuário `default` que tem uma string de acesso modificada.

Para modificar o usuário padrão no console

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Escolha Gerenciamento de grupos de usuários no painel de navegação.

3. Em ID do grupo de usuários, escolha o ID que você deseja modificar. Verifique se você escolheu o link e não a caixa de seleção.
4. Escolha Modificar.
5. Na janela Modificar, escolha Gerenciar e selecione o usuário que você deseja como o usuário padrão com o Nome do usuário como o padrão.
6. Selecione Choose (Escolher).
7. Escolha Modificar. Quando você fizer isso, todas as conexões existentes com um cache que o usuário padrão original tiver serão encerradas.

Para modificar o usuário padrão com a AWS CLI

1. Criar um novo usuário com o nome de usuário default usando os seguintes comandos.

Para Linux, macOS ou Unix:

```
aws elasticache create-user \  
  --user-id "new-default-user" \  
  --user-name "default" \  
  --engine "REDIS" \  
  --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 destas restrições de senha de usuário ao usar o RBAC com ElastiCache para Redis:

- As senhas devem ter de 16 a 128 caracteres imprimíveis.
- Os seguintes caracteres não alfanuméricos não são permitidos: , " " / @.

Gerenciamento de usuários com o console e a CLI

Use o procedimento a seguir para gerenciar usuários no console.

Para gerenciar usuários no console

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel do Amazon ElastiCache, escolha Gerenciamento de usuários. As seguintes opções estão disponíveis:
 - Criar usuário: ao criar um usuário, você insere um ID de usuário, um nome de usuário, um modo de autenticação e uma string de acesso. A string de acesso define o nível de permissão para quais chaves e comandos o usuário é permitido.

Ao criar um usuário, você pode configurar até duas senhas. Quando você modifica uma senha, todas as conexões existentes com os caches são mantidas.

- Modificar usuário: permite a você atualizar as configurações de um usuário ou alterar a string de acesso.
- Excluir usuário: a conta de usuário será removida de todos os grupos de usuários aos quais ela pertence.

Use o procedimento a seguir para gerenciar usuários na AWS CLI.

Para modificar um usuário usando a CLI

- Use o comando `modify-user` para atualizar a senha ou senhas de um usuário ou alterar as permissões de acesso de um usuário.

Quando um usuário é modificado, os grupos de usuários associados ao usuário são atualizados com todos os caches associados ao grupo de usuários. Todas as conexões existentes são mantidas. Veja os exemplos a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-user \  
  --user-id user-id-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --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 a CLI

- Use o comando `delete-user` para excluir um usuário. A conta é excluída e removida de todos os grupos de usuários aos quais pertence. Veja um exemplo a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache delete-user \  
--user-id user-id-2
```

Para Windows:

```
aws elasticache delete-user ^  
--user-id user-id-2
```

Para ver uma lista de usuários, execute a operação [describe-users](#).

```
aws elasticache describe-users
```

Gerenciamento de grupos de usuários com o console e a CLI

É possível criar grupos de usuários para organizar e controlar o acesso de usuários a um ou mais caches, conforme mostrado a seguir.

Use o procedimento a seguir para gerenciar grupos de usuários no console.

Como gerenciar grupos de usuários usando o console

1. Faça login no AWS Management Console e abra o console do Amazon ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel Amazon ElastiCache, 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 na CLI.

Para criar um novo grupo de usuários e adicionar um usuário usando a CLI

- Use o comando `create-user-group`, conforme mostrado a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache create-user-group \  
  --user-group-id "new-group-1" \  
  --engine "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 a CLI

- Use o comando `modify-user-group`, conforme mostrado a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-user-group --user-group-id new-group-1 \  
  --user-ids-to-add user-id-3 \  
  --user-ids-to-remove user-id-2
```

Para Windows:

```
aws elasticache modify-user-group --user-group-id new-group-1 ^  
  --user-ids-to-add 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 a CLI

- Use o comando `delete-user-group`, conforme mostrado a seguir. O grupo de usuários em si, não os usuários pertencentes ao grupo, é excluído.

Para Linux, macOS ou Unix:

```
aws elasticache delete-user-group /  
  --user-group-id
```

Para Windows:

```
aws elasticache delete-user-group ^  
  --user-group-id
```

Para ver uma lista de grupos de usuários, execute a operação [describe-user-groups](#).

```
aws elasticache describe-user-groups \  
  --user-group-id test-group
```

Atribuição de grupos de usuários a caches sem servidor

Depois que você tiver criado um grupo de usuários e adicionado usuários, a etapa final na implementação do RBAC será atribuir o grupo de usuários a um cache sem servidor.

Atribuição de grupos de usuários a caches sem servidor usando o console

Para adicionar um grupo de usuários a cache sem servidor usando o AWS Management Console, faça o seguinte:

- Para o modo cluster desabilitado, consulte [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- Para o modo cluster habilitado, consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)

Atribuição de grupos de usuários a caches sem servidor usando o console AWS CLI

A operação da AWS CLI a seguir cria um cache sem servidor usando o parâmetro `user-group-id` com o valor *my-user-group-id*. Substitua o grupo de sub-rede `sng-test` por um grupo de sub-rede que exista.

Principais parâmetros

- **--engine**: deve ser `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 operação da AWS CLI a seguir modifica um cache sem servidor com o parâmetro `user-group-id` 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 [Visualização de eventos do ElastiCache](#).

Atribuição de grupos de usuários a grupos de replicação

Depois de criar um grupo de usuários e adicionar usuários, a etapa final na implementação do RBAC é atribuir o grupo de usuários a um grupo de replicação.

Atribuição de grupos de usuários a grupos de replicação usando o console

Para adicionar um grupo de usuários a uma replicação usando o AWS Management Console, faça o seguinte:

- Para o modo cluster desabilitado, consulte [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#)
- Para o modo cluster habilitado, consulte [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#)

Atribuição de grupos de usuários a grupos de replicação usando a AWS CLI

A operação da AWS CLI a seguir cria um grupo de replicação com criptografia em trânsito (TLS) habilitada e o parâmetro `user-group-ids` com o valor `my-user-group-id`. Substitua o grupo de sub-rede `sng-test` por um grupo de sub-rede que exista.

Principais parâmetros

- **--engine**: deve ser `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 seguinte operação da AWS CLI modifica um grupo de replicação com criptografia em trânsito (TLS) habilitada e o parâmetro `user-group-ids` 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 [Visualização de eventos do ElastiCache](#).

Migração do Redis AUTH para o RBAC

Se você estiver usando o Redis AUTH conforme descrito em [Autenticar com o comando Redis AUTH](#) e deseja migrar para usar o RBAC, use os seguintes procedimentos.

Use o procedimento a seguir para migrar do Redis AUTH para o RBAC usando o console.

Para migrar do Redis AUTH para o RBAC usando o console

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista no canto superior direito, escolha a região da AWS onde 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 Redis AUTH para o RBAC usando a CLI.

Para migrar do Redis AUTH para o RBAC usando a CLI

- Use o comando `modify-replication-group`, conforme mostrado a seguir.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group --replication-group-id test \  
  --auth-token-update-strategy DELETE \  
  --user-group-ids-to-add user-group-1
```

Para Windows:

```
aws elasticache modify-replication-group --replication-group-id test ^  
--auth-token-update-strategy DELETE ^  
--user-group-ids-to-add user-group-1
```

Migração do RBAC para o Redis AUTH

Se você estiver usando o RBAC e desejar migrar para o Redis AUTH, consulte [Migração do RBAC para o Redis AUTH](#).

Note

Se precisar desabilitar o controle de acesso em um cache do ElastiCache, você precisará fazer isso por meio da AWS CLI. Para obter mais informações, consulte [the section called “Desabilitar o controle de acesso em um cache do ElastiCache para Redis”](#).

Alternância automática de senhas para os usuários

Com o AWS Secrets Manager, é possível substituir credenciais codificadas, incluindo senhas, por uma chamada de API ao Secrets Manager para recuperar o segredo por programação. 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 as senhas do ElastiCache para Redis (ou seja, os segredos) usando uma função do AWS Lambda fornecida pelo Secrets Manager.

Para obter mais informações sobre AWS Secrets Manager, consulte [O que é AWS Secrets Manager?](#)

Como o ElastiCache usa segredos

Com o Redis 6, o ElastiCache para Redis apresentou o [Regras de controle de acesso com base em função \(RBAC\)](#) para proteger o cluster do Redis. Esse recurso permite que certas conexões sejam limitadas em termos dos comandos que podem ser executados e das chaves que podem

ser acessadas. Com o RBAC, enquanto o cliente cria um usuário com senhas, os valores da senha precisam ser inseridos manualmente em texto sem formatação e ficam visíveis para o operador.

Com o Secrets Manager, as aplicações buscam a senha do Secrets Manager em vez de inseri-la manualmente e armazená-la na configuração da aplicação. Para obter informações sobre como fazer isso, consulte [Como os usuários do ElastiCache são associados ao segredo](#).

O uso de segredos tem um custo. Para obter informações sobre preços, consulte [Preços do AWS](#).

Como os usuários do ElastiCache são associados ao segredo

O Secrets Manager manterá uma referência para o usuário associado no campo SecretString do segredo. Não haverá referência ao segredo no lado do ElastiCache.

```
{
  "password": "strongpassword",
  "username": "user1",
  "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1" //this is the
  bond between the secret and the user
}
```

Função de alternância do Lambda

Para habilitar a alternância automática de senhas do Secrets Manager, você criará uma função do Lambda que interagirá com a API [modify-user](#) para atualizar as senhas do usuário.

Para obter informações sobre como isso funciona, consulte [Como funciona a alternância](#).

Note

Para alguns produtos da AWS, para evitar o cenário de “confused deputy”, a AWS recomenda usar as chaves de condição globais `aws:SourceArn` e `aws:SourceAccount`. No entanto, se você incluir a condição `aws:SourceArn` em sua política de função de alternância, a função de alternância só poderá ser usada para alternar o segredo especificado por esse ARN. Recomendamos que inclua apenas a chave de contexto `aws:SourceAccount`, de modo que possa usar a função de alternância para vários segredos.

Para qualquer problema que você possa encontrar, consulte [Resolução de problemas AWS Secrets Manager](#).

Como criar um usuário do ElastiCache 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 do 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 Save (Salvar).
- f. Escolha Environment variables (Variáveis de ambiente) e defina o seguinte:

- i. SECRETS_MANAGER_ENDPOINT – <https://secretsmanager>.
REGION.amazonaws.com
 - ii. SECRET_ARN – O nome de recurso da Amazon (ARN) do segredo que você criou na etapa 2.
 - iii. USER_NAME – Nome de usuário do usuário do ElastiCache,
 - iv. Escolha Save (Salvar).
- g. Escolha Permissions (Permissões)
 - h. Em Execution role (Função de execução), escolha o nome da função do Lambda para exibição no console do IAM.
 - i. A função do Lambda precisará da seguinte permissão para modificar os usuários e definir a senha:

ElastiCache

```
{
  "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:xxxxxxxxxxxx:secret:XXXX"
  },
  {
    "Effect": "Allow",
    "Action": "secretsmanager:GetRandomPassword",
    "Resource": "*"
  }
]
```

4. Configurar a alternância de segredos do Secrets Manager

- a. Usando o AWS Management Console, consulte [Configuração da alternância automática de segredos do AWS 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 alternância automática para o AWS Secrets Manager usando o AWS Command Line Interface](#)

Autenticação com o IAM

Tópicos

- [Visão geral](#)
- [Limitações](#)
- [Configuração](#)
- [Conexão](#)

Visão geral

Com a autenticação do IAM, é possível autenticar uma conexão com o ElastiCache para Redis usando identidades do AWS IAM, quando o cache está configurado para usar a versão 7 ou posterior. Isso possibilita que você fortaleça seu modelo de segurança e simplifique várias tarefas administrativas de segurança. Você também pode usar a autenticação do IAM para configurar um controle de acesso refinado para cada cache do ElastiCache individual e usuário do ElastiCache

seguindo os princípios das permissões de privilégio mínimo. A autenticação do IAM para o ElastiCache para Redis funciona fornecendo um token de autenticação do IAM de curta duração, em vez de uma senha de usuário do ElastiCache de longa duração no comando AUTH ou HELLO do Redis. Para obter mais informações sobre o token de autenticação do IAM, consulte o [processo de assinatura do Signature versão 4](#) no Guia de referência geral do AWS e no exemplo de código abaixo.

Você pode usar as identidades do IAM e suas políticas associadas para restringir ainda mais o acesso ao Redis. Também é possível conceder acesso a usuários dos provedores de identidade federados diretamente para caches do Redis.

Para usar o AWS IAM com o ElastiCache para Redis, primeiro é necessário criar um usuário do ElastiCache com o modo de autenticação definido como IAM e criar ou reutilizar uma identidade do IAM. A identidade do IAM precisa de uma política associada para conceder a ação `elasticache:Connect` ao cache do ElastiCache e ao usuário do ElastiCache. Depois de configurado, você pode criar um token de autenticação do IAM usando as credenciais da AWS do usuário ou do perfil do IAM. Por fim, você precisa fornecer o token de autenticação do IAM de curta duração como uma senha no cliente do Redis ao se conectar ao cache do Redis. Um cliente Redis com suporte para provedor de credenciais pode gerar automaticamente as credenciais temporárias para cada nova conexão. O ElastiCache para Redis executará a autenticação do IAM para solicitações de conexão de usuários do ElastiCache habilitados para o IAM e validará as solicitações de conexão com o IAM.

Limitações

Ao usar a autenticação do IAM, as seguintes limitações se aplicam:

- A autenticação do IAM está disponível ao usar o ElastiCache para Redis versão 7.0 ou posterior.
- Para usuários do ElastiCache habilitados para o IAM, as propriedades de nome de usuário e ID de usuário devem ser idênticas.
- O token de autenticação do IAM é válido por 15 minutos. Para conexões de longa duração, recomendamos usar um cliente Redis que ofereça suporte a uma interface de provedor de credenciais.
- Uma conexão autenticada pelo IAM com o ElastiCache para Redis será automaticamente desconectada após 12 horas. A conexão pode ser prolongada por 12 horas enviando um comando AUTH ou HELLO com um novo token de autenticação do IAM.
- A autenticação do IAM não tem suporte em comandos MULTI EXEC.

- Atualmente, a autenticação do IAM não dá suporte às seguintes chaves de contexto de condição global:
 - Durante o uso da autenticação do IAM com caches sem servidor, `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` e `aws:ResourceTag/%s` (de caches sem servidor associados e usuários) são compatíveis.
 - Durante o uso da autenticação do IAM com grupos de replicação, `aws:SourceIp` e `aws:ResourceTag/%s` (de grupos de replicação associados e usuários) são compatíveis.

Para obter mais informações sobre chaves de contexto de condição global, consulte [Chaves de contexto de condição global da AWS](#) no Guia do usuário do IAM.

Configuração

Como configurar a autenticação do IAM

1. Criar um cache

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

2. Crie um documento de política de confiança do IAM, conforme mostrado abaixo, para o perfil que permita que sua conta assumo o novo perfil. 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 de política do IAM, 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. Criar um perfil do IAM.

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

5. Crie a política do IAM.

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

6. Anexe a política do IAM à função.

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

7. Crie um novo usuário habilitado para o IAM.

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

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

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

Conexão

Conectar com token como senha

Primeiro, é necessário gerar o token de autenticação do IAM de curta duração usando uma [solicitação pré-assinada do AWS SigV4](#). Depois disso, você fornecerá o token de autenticação do IAM como senha ao se conectar a um cache do 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.  
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,  
  region, isServerless);  
String iamAuthToken =  
  iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());  
  
// Construct Redis 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 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 autenticar com o ElastiCache para Redis usando o provedor de credenciais de autenticação do IAM.

```

String userId = "insert user id";
String cacheName = "insert cache name";

```

```
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
    DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request. Once this request is signed it can be
// used as an
// IAM authentication token for ElastiCache Redis.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);

// Create a Redis credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
    RedisIAMAuthCredentialsProvider(
        userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
    .build();

// Create a new Lettuce Redis client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Veja abaixo um exemplo de um cliente Lettuce Redis que encapsula o `IAMAuthTokenRequest` em um provedor de credenciais para 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 = userName;
    this.awsCredentialsProvider = awsCredentialsProvider;
    this.iamAuthTokenRequest = iamAuthTokenRequest;
    this.iamAuthTokenSupplier =
Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
    TimeUnit.SECONDS);
}

@Override
public Mono<RedisCredentials> resolveCredentials() {
    return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));
}

private String getIamAuthToken() {
    return
iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
}
}
```

Autenticar com o comando Redis AUTH

Note

O Redis AUTH foi substituído por [the section called “Regras de controle de acesso com base em função \(RBAC\)”](#). Todos os caches sem servidor devem usar o RBAC na autenticação.

Os tokens de autenticação do Redis, ou senhas, habilitam o Redis a exigir uma senha antes de permitir que os clientes executem comandos, melhorando assim a segurança dos dados. O Redis AUTH só está disponível para clusters autoprojettados.

Tópicos

- [Visão geral do AUTH no ElastiCache para Redis](#)
- [Aplicação de autenticação a um cluster do ElastiCache para Redis](#)
- [Modificação do token AUTH em um cluster existente do ElastiCache para Redis](#)
- [Migração do RBAC para o Redis AUTH](#)

Visão geral do AUTH no ElastiCache para Redis

Quando você utiliza o Redis AUTH com seu cluster do ElastiCache para Redis, há alguns refinamentos.

Em particular, esteja ciente destas restrições de token, ou senha de AUTH, ao usar AUTH com ElastiCache para Redis:

- Tokens, ou senhas, devem ter de 16 a 128 caracteres imprimíveis.
- Caracteres não alfanuméricos são restritos a (!, &, #, \$, ^, <, >, -).
- O AUTH só pode ser habilitado para criptografia em trânsito para clusters do ElastiCache para Redis.

Para configurar um token forte, recomendamos que você siga uma política de senha estrita, tal como exigir o seguinte:

- Tokens, ou senhas, devem incluir pelo menos três dos seguintes tipos de caractere:
 - Caracteres maiúsculos
 - Caracteres minúsculos
 - Dígitos
 - Caracteres não alfanuméricos (!, &, #, \$, ^, <, >, -)
- Tokens, ou senhas, não devem conter uma palavra de dicionário ou uma palavra de dicionário minimamente alterada.
- Tokens, ou senhas, não devem ser iguais ou similares a um token usado recentemente.

Aplicação de autenticação a um cluster do ElastiCache para Redis

Você pode exigir que os usuários insiram um token (senha) em um servidor Redis protegido por tokens. Para fazer isso, inclua o parâmetro `--auth-token` (API: `AuthToken`) com o token correto quando você criar seu grupo de replicação ou cluster. Inclua-o também em todos os comandos subsequentes para o grupo de replicação ou cluster.

A operação da AWS CLI a seguir cria um grupo de replicação com criptografia em trânsito (TLS) habilitada e o token AUTH *This-is-a-sample-token*. Substitua o grupo de sub-rede `sng-test` por um grupo de sub-rede que exista.

Principais parâmetros

- **--engine**: deve ser `redis`.
- **--engine-version**: deve ser 3.2.6, 4.0.10 ou posterior.
- **--transit-encryption-enabled**: obrigatório para autenticação e qualificação para HIPAA.
- **--auth-token**: obrigatório para qualificação para HIPAA. Esse valor deve ser o token correto para esse servidor Redis protegido por tokens.
- **--cache-subnet-group**: obrigatório para qualificação para HIPAA.

Para Linux, macOS ou Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id authtestgroup \  
  --replication-group-description authtest \  
  --engine redis \  
  --cache-node-type cache.m4.large \  
  --num-node-groups 1 \  
  --replicas-per-node-group 2 \  
  --transit-encryption-enabled \  
  --auth-token This-is-a-sample-token \  
  --cache-subnet-group sng-test
```

Para Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id authtestgroup ^  
  --replication-group-description authtest ^  
  --engine redis ^  
  --cache-node-type cache.m4.large ^  
  --num-node-groups 1 ^  
  --replicas-per-node-group 2 ^  
  --transit-encryption-enabled ^  
  --auth-token This-is-a-sample-token ^  
  --cache-subnet-group sng-test
```

Modificação do token AUTH em um cluster existente do ElastiCache para Redis

Para facilitar a atualização da autenticação, você pode modificar o token AUTH usado em um cluster do ElastiCache para Redis. Você poderá fazer essa modificação se a versão do mecanismo for 5.0.6 ou superior e se o ElastiCache para Redis tiver criptografia em trânsito habilitada.

A modificação do token de autenticação oferece suporte a duas estratégias: ROTATE (ALTERNAR) e SET (DEFINIR). A estratégia ROTATE (ALTERNAR) acrescenta um token AUTH adicional para o servidor enquanto mantém o token anterior. A estratégia SET (DEFINIR) atualiza o servidor para suportar apenas um único token AUTH. Faça essas chamadas de modificação com o parâmetro `--apply-immediately` para aplicar as alterações imediatamente.

Alternância do token AUTH

Para atualizar um servidor Redis com um novo token AUTH, execute a API `ModifyReplicationGroup` com o parâmetro `--auth-token` como o novo token de autenticação e o parâmetro `--auth-token-update-strategy` com o valor ROTATE (ALTERNAR). Depois que a modificação for concluída, o cluster oferecerá suporte ao token AUTH anterior além do especificado no parâmetro `auth-token`.

Note

Se você não configurar o token AUTH antes, depois que a modificação for concluída, o cluster não oferecerá suporte a nenhum token AUTH além do especificado no parâmetro `auth-token`.

Se essa modificação for executada em um servidor que já ofereça suporte a dois tokens AUTH, o token AUTH mais antigo também será removido durante esta operação, permitindo que um servidor suporte até dois tokens AUTH mais recentes em um determinado momento.

Neste ponto, você pode continuar atualizando o cliente para usar o token AUTH mais recente. Depois que os clientes forem atualizados, você poderá usar a estratégia SET (DEFINIR) para alternância de token AUTH (explicada na seção a seguir) para começar exclusivamente a usar o novo token.

A operação da AWS CLI a seguir modifica um grupo de replicação para alternar o token AUTH ***This-is-the-rotated-token.***

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-rotated-token \  
--auth-token-update-strategy ROTATE \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-rotated-token ^  
--auth-token-update-strategy ROTATE ^  
--apply-immediately
```

Definição do token AUTH

Para atualizar um servidor Redis com dois tokens AUTH para o suporte a um único token AUTH, execute a operação da API `ModifyReplicationGroup`. Execute `ModifyReplicationGroup` com o parâmetro `--auth-token` como o novo token AUTH e o parâmetro `--auth-token-update-strategy` com o valor SET (DEFINIR). O parâmetro `auth-token` deve ser o mesmo valor do último token AUTH alternado. Após a conclusão da modificação, o servidor Redis oferecerá suporte apenas para o token AUTH especificado no parâmetro `auth-token`.

A operação da AWS CLI a seguir modifica um grupo de replicação para definir o token AUTH como *This-is-the-set-token*.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-set-token \  
--auth-token-update-strategy SET \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-set-token ^  
--auth-token-update-strategy SET ^  
--apply-immediately
```

Habilitação de autenticação em um cluster existente do ElastiCache para Redis

Para habilitar a autenticação em um servidor do Redis existente, execute a operação da API `ModifyReplicationGroup`. Execute `ModifyReplicationGroup` com o parâmetro `--auth-`

token como o novo token e o parâmetro `--auth-token-update-strategy` com o valor `ROTATE` (ALTERNAR).

Depois que a modificação for concluída, o cluster oferecerá suporte ao token AUTH especificado no parâmetro `auth-token`, além do suporte a conexões sem autenticação. A habilitação da autenticação só tem suporte para servidores Redis com criptografia em trânsito (TLS) habilitada.

Migração do RBAC para o Redis AUTH

Se você estiver autenticando usuários com o controle de acesso baseado em função do Redis (RBAC), conforme descrito em [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) e desejar migrar para o Redis AUTH, use os seguintes procedimentos. Você pode migrar usando o console ou a CLI.

Para migrar do RBAC para o Redis AUTH usando o console

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista localizada no canto superior direito, escolha a região da AWS em que o cluster que você deseja modificar está localizado.
3. No painel de navegação, escolha o mecanismo em execução no cluster que deseja modificar.
É exibida uma lista dos clusters do mecanismo escolhido.
4. Na lista de clusters, no cluster que você deseja modificar, escolha seu nome.
5. Para Actions (Ações), escolha Modify (Modificar).

A janela Modificar é exibida.

6. Para Opção de controle de acesso, escolha Acesso do usuário padrão AUTH do Redis.
7. Em Token AUTH do Redis, defina um novo token.
8. Escolha Previsualizar alterações e, na próxima tela, Modificar.

Para migrar do RBAC para o Redis AUTH usando a AWS CLI

- Use um dos seguintes comandos.

Para Linux, macOS ou Unix:

```
aws elasticache modify-replication-group \
```

```
--replication-group-id test \  
--remove-user-groups \  
--auth-token password \  
--auth-token-update-strategy SET \  
--apply-immediately
```

Para Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id test ^  
--remove-user-groups ^  
--auth-token password ^  
--auth-token-update-strategy SET ^  
--apply-immediately
```

Para obter mais informações sobre como trabalhar com o AUTH, consulte [Token AUTH](#) no site [redis.io](#).

Note

Se você precisar desabilitar o controle de acesso em um cluster do ElastiCache, consulte [the section called “Desabilitar o controle de acesso em um cache do ElastiCache para Redis”](#).

Desabilitar o controle de acesso em um cache do ElastiCache para Redis

Siga as instruções abaixo para desabilitar o controle de acesso em um cache do Redis habilitado para TLS. Seu cache do Redis terá um dos dois tipos diferentes de configurações: acesso de usuário padrão AUTH do Redis ou lista de controle de acesso de grupo de usuários (RBAC). Se o cache tiver sido criado com a configuração AUTH, você precisará alterá-lo para a configuração RBAC antes de poder desabilitar o cache por meio da remoção dos grupos de usuários. Se o cache tiver sido criado com a configuração RBAC, você poderá desabilitá-lo diretamente.

Como desabilitar um cache sem servidor do 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": ""
  "...
}
```

Como desabilitar um cache do Redis configurado com um token AUTH

1. Altere o token AUTH para RBAC e especifique um grupo de usuários a ser adicionado.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-value>
```

2. Verifique se o token AUTH foi desabilitado e se um grupo de usuários foi adicionado.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-id-value>
{
  "...
  "AuthTokenEnabled": false,
  "UserGroupIds": [
    "<user-group-value>"
  ]
  "...
}
```

3. Remova os grupos de usuários para desabilitar o controle de acesso.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
  "...
  "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
```

```

        "<user-group-value>"
    ]
}
"..."
}

```

4. (Opcional) Verifique se nenhum grupo de usuários está associado ao cluster. O campo `AuthTokenEnabled` também deve ser falso.

```

aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false

```

Como desabilitar um cluster do 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

O Amazon ElastiCache usa as seguintes técnicas para proteger seus dados de cache e protegê-los contra o acesso não autorizado:

- [Segurança do Amazon VPCs e do ElastiCache](#) explica o tipo de grupo de segurança de que você precisa para sua instalação.
- [Gerenciamento de identidade e acesso no Amazon ElastiCache](#) para conceder e limitar ações de usuários, grupos e funções.

Segurança do Amazon VPCs e do ElastiCache

Como a segurança dos dados é importante, o ElastiCache fornece meios para você controlar quem tem acesso aos seus dados. A forma como você controla o acesso aos seus dados depende de você ter ou não executado seus clusters em uma Amazon Virtual Private Cloud (Amazon VPC) ou Amazon EC2-Classic.

Important

Tornamos o uso do Amazon EC2-Classic obsoleto para iniciar clusters do ElastiCache. Todos os nós de geração atuais são iniciados somente na Amazon Virtual Private Cloud.

O serviço da Amazon Virtual Private Cloud (Amazon VPC) define uma rede virtual que lembra muito um datacenter tradicional. Ao configurar sua Amazon VPC, você pode selecionar o intervalo de endereços IP dela, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações de segurança. Você também pode adicionar um cluster de cache à rede virtual e controlar o acesso a esse cluster usando grupos de segurança da Amazon VPC.

Esta seção explica como configurar manualmente um cluster do ElastiCache em uma Amazon VPC. Essas informações destinam-se a usuários que desejam uma compreensão mais profunda de como o ElastiCache e a Amazon VPC trabalham juntas.

Tópicos

- [Entendendo o ElastiCache e as Amazon VPCs](#)
- [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#)
- [Criar uma nuvem privada virtual \(VPC\)](#)

- [Conexão a um cache em execução em um Amazon VPC](#)

Entendendo o ElastiCache e as Amazon VPCs

O ElastiCache está totalmente integrado à Amazon Virtual Private Cloud (Amazon VPC). Para usuários do ElastiCache, isso significa o seguinte:

- Se a sua conta da AWS oferece suporte apenas para a plataforma EC2-VPC, o ElastiCache sempre inicia seu cluster em uma Amazon VPC.
- Se você for iniciante na AWS, seus clusters serão implantados em uma Amazon VPC. Uma VPC padrão será criada para você automaticamente.
- Se você tiver uma VPC padrão e não especificar uma sub-rede quando executar um cluster, este será iniciado na sua Amazon VPC padrão.

Para mais informações, consulte [Detecção de suas plataformas compatíveis e se você tem um VPC padrão](#).

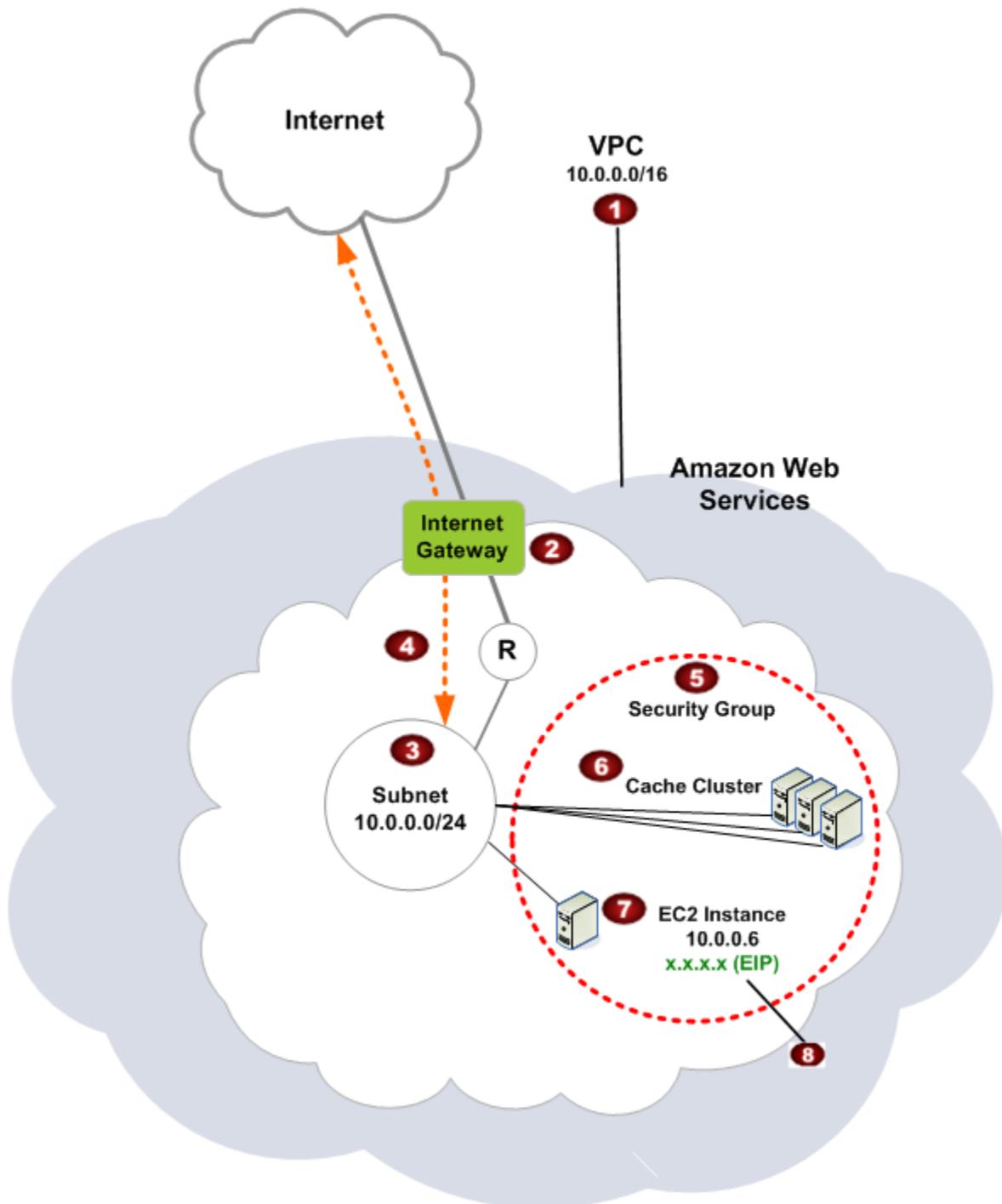
Com a Amazon Virtual Private Cloud, você pode criar uma rede virtual na nuvem da AWS que se assemelha muito com um datacenter tradicional. É possível configurar sua Amazon VPC, incluindo selecionar o intervalo de endereços IP, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações de segurança.

A funcionalidade básica do ElastiCache é a mesma em uma nuvem privada virtual. O ElastiCache gerencia atualizações de software, a aplicação de patch, a detecção de falhas e a recuperação, independentemente de os seus clusters estarem implantados dentro ou fora de uma Amazon VPC.

Os nós de cache do ElastiCache implantados fora de uma Amazon VPC recebem um endereço IP para o qual o endpoint/nome do DNS é resolvido. Isso fornece conectividade entre as instâncias do Amazon Elastic Compute Cloud (Amazon EC2). Quando você ativa um cluster do ElastiCache na sub-rede privada de uma Amazon VPC, cada nó de cache recebe um endereço IP privado dentro dessa sub-rede.

Visão geral do ElastiCache em uma Amazon VPC

O seguinte diagrama e tabela descrevem o ambiente da Amazon VPC, juntamente com os clusters do ElastiCache e as instâncias do Amazon EC2 que são executadas na Amazon VPC.

**1**

A Amazon VPC é uma parte isolada da nuvem da AWS que recebe seu próprio bloco de endereços IP.

2

Um gateway da Internet conecta sua Amazon VPC diretamente à Internet e fornece acesso a outros recursos da AWS, como o Amazon Simple Storage Service (Amazon S3), que estão em execução fora da sua Amazon VPC.

3

Uma sub-rede da Amazon VPC é um segmento do intervalo de endereços IP de uma Amazon VPC em que você pode isolar recursos da AWS de acordo com suas necessidades de segurança e operacionais.

4

Uma tabela de roteamento na Amazon VPC direciona o tráfego de rede entre a sub-rede e a Internet. A Amazon VPC tem um roteador implícito, que é simbolizado neste diagrama pelo círculo com o R.

5

Um grupo de segurança da Amazon VPC controla o tráfego de entrada e saída para seus clusters do ElastiCache e suas instâncias do Amazon EC2.

6

Você pode ativar um cluster do ElastiCache na sub-rede. Os nós de cache possuem endereços IP privados a partir do intervalo de endereços da sub-rede.

7

Você também pode ativar instâncias do Amazon EC2 na sub-rede. Cada instância do Amazon EC2 tem um endereço IP privado do intervalo de endereços da sub-rede. A instância do Amazon EC2 pode se conectar a qualquer nó de cache na mesma sub-rede.

8

Para que uma instância do Amazon EC2 na sua Amazon VPC seja acessível na Internet, você precisa atribuir um endereço público estático chamado de endereço IP elástico à instância.

Pré-requisitos

Para criar um cluster do ElastiCache dentro de uma Amazon VPC, sua Amazon VPC deve atender aos seguintes requisitos:

- A Amazon VPC deve permitir instâncias do Amazon EC2 não dedicadas. Você não pode usar o ElastiCache em uma Amazon VPC que está configurada para a locação de instâncias dedicadas.

- Um grupo de sub-redes de cache deve ser definido para a sua Amazon VPC. O ElastiCache usa esse grupo de sub-redes de cache para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos endpoints da VPC ou aos nós de cache.
- Os blocos CIDR para cada sub-rede devem ser suficientemente grandes para fornecer endereços IP de reposição para o ElastiCache usar durante atividades de manutenção.

Roteamento e segurança

Você pode configurar o roteamento na sua Amazon VPC para controlar para onde o tráfego flui (por exemplo, para o gateway da Internet ou o virtual private gateway). Com um gateway da Internet, sua Amazon VPC tem acesso direto a outros recursos da AWS que não estão sendo executados na sua Amazon VPC. Se você optar por ter apenas um gateway privado virtual com uma conexão com a rede local da sua organização, poderá rotear seu tráfego vinculado à Internet através da VPN e usar políticas de segurança locais e um firewall para controlar a saída. Nesse caso, você está sujeito a cobranças adicionais de largura de banda ao acessar os recursos da AWS pela Internet.

Você pode usar grupos de segurança do Amazon VPC para ajudar a proteger os clusters do ElastiCache e as instâncias do Amazon EC2 na sua Amazon VPC. Os security groups atuam como um firewall no nível da instância e não no nível da sub-rede.

Note

Recomendamos enfaticamente que você use nomes DNS para se conectar aos seus nós, pois o endereço IP subjacente pode mudar.

Documentação da Amazon VPC

A Amazon VPC tem seu próprio conjunto de documentação para descrever como criar e usar sua Amazon VPC. A tabela a seguir fornece links para guias sobre Amazon VPCs.

Descrição	Documentação
Como começar a usar a Amazon VPC	Conceitos básicos do Amazon VPC
Como usar a Amazon VPC por meio do AWS Management Console	Guia do usuário da Amazon VPC

Descrição	Documentação
Descrições completas de todos os comandos da Amazon VPC	Referência da linha de comando do Amazon EC2 (os comandos da Amazon VPC são encontrados na referência do Amazon EC2)
Descrições completas das operações da API da Amazon VPC, tipos de dados e erros da Amazon VPC	Referência da API do Amazon EC2 (as operações da Amazon VPC são encontrados na referência do Amazon EC2)
Informações para o administrador da rede que precisa configurar o gateway no final de uma conexão VPN IPsec opcional	O que é a AWS VPN Site-to-Site?

Para obter informações mais detalhadas sobre a Amazon Virtual Private Cloud, consulte [Amazon Virtual Private Cloud](#).

Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC

O Amazon ElastiCache oferece suporte para os seguintes cenários para acessar um cache em um Amazon VPC:

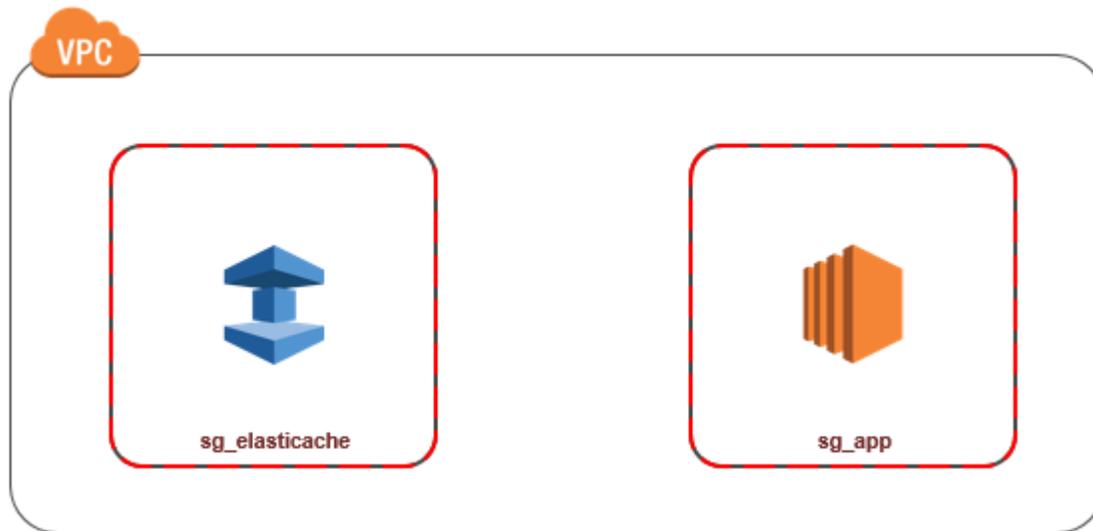
Sumário

- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão no mesmo Amazon VPC](#)
- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs](#)
 - [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs na mesma região](#)
 - [Uso do Transit Gateway](#)
 - [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs em regiões diferentes](#)
 - [Uso da VPC de trânsito](#)
- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente](#)
 - [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando conectividade de VPN](#)
 - [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando o Direct Connect](#)

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão no mesmo Amazon VPC

O caso de uso mais comum é quando uma aplicação implantada em uma instância do EC2 precisa se conectar a um cache na mesma VPC.

O diagrama a seguir ilustra esse cenário



A maneira mais simples de gerenciar o acesso entre instâncias do EC2 e caches na mesma VPC é fazer o seguinte:

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

A porta padrão para os caches Redis é 6379.

2. Crie um grupo de segurança de VPC para suas instâncias do EC2 (servidores Web e de aplicativos). Esse grupo de segurança pode, se necessário, permitir o acesso à instância do EC2 da Internet através da tabela de rotas da VPC. Por exemplo, você pode definir regras nesse grupo de segurança para permitir o acesso TCP à instância do EC2 pela porta 22.
3. Crie regras personalizadas no grupo de segurança para o seu cache que permitam conexões do grupo de segurança que você criou para suas instâncias do EC2. Isso permitiria que qualquer membro de grupo de segurança acessasse os caches.

Note

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

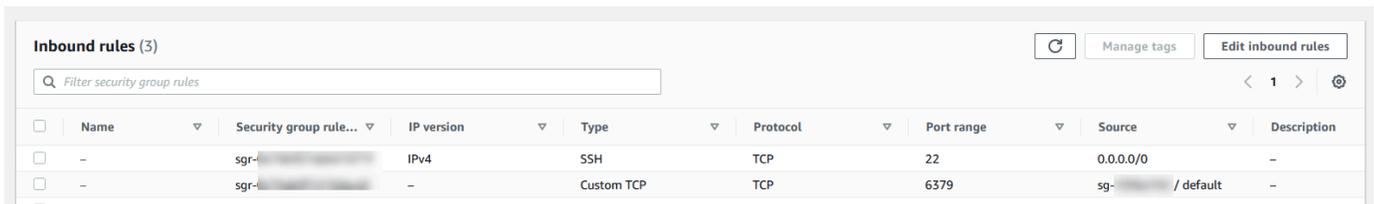
de disponibilidade. Todos os gateways e tabelas de rotas relevantes serão ajustados automaticamente.

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

1. Faça login no Console de Gerenciamento da AWS e abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc>.
2. No painel de navegação, selecione Grupos de segurança.
3. Selecione ou crie um grupo de segurança que você usará para seus caches. Em Regras de entrada, selecione Editar regras de entrada e escolha Adicionar regra. Esse grupo de segurança permitirá o acesso a membros de outro grupo de segurança.
4. Em Tipo, escolha Regra TCP personalizada.
 - a. Para Port Range, especifique a porta que você usou quando criou seu cache.

A porta padrão dos caches e grupos de replicação 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 o suas instâncias do Amazon EC2.
5. Escolha Save quando terminar.



	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	-

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs

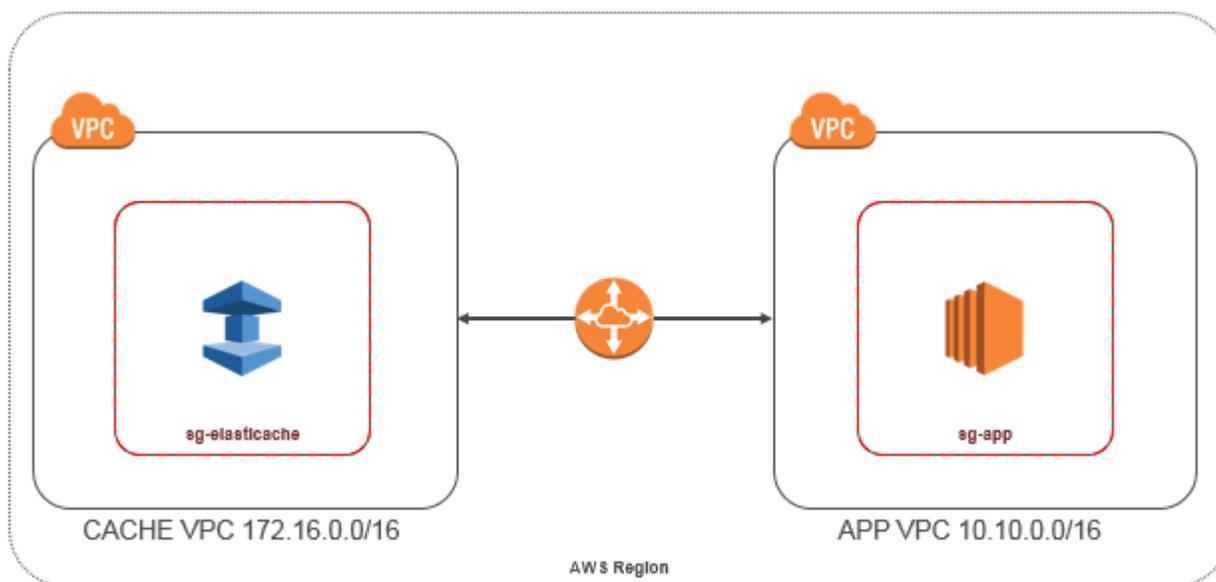
Quando seu cache está em uma VPC diferente da instância do EC2 que você está usando para acessá-la, existem várias maneiras de acessar o cache. Se o cache e a instância do EC2 estiverem em VPCs diferentes, mas na mesma região, você poderá usar o emparelhamento de VPCs. Se o cache e a instância do EC2 estiverem em regiões diferentes, você poderá criar conectividade via VPN entre regiões.

Tópicos

- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs na mesma região](#)
- [Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs em regiões diferentes](#)

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs na mesma região

O diagrama a seguir ilustra o acesso a um cache por uma instância do Amazon EC2 em um Amazon VPC diferente na mesma região usando uma conexão de emparelhamento do Amazon VPC.



Cache acessado por uma instância do Amazon EC2 em um Amazon VPC diferente na mesma região: conexão de emparelhamento da VPC

Uma conexão de emparelhamento da VPC é uma conexão de redes entre duas VPCs que permite direcionar o tráfego entre elas usando endereços IP privados. Instâncias em qualquer VPC podem se comunicar umas com as outras como se estivessem na mesma rede. Você pode criar uma conexão de emparelhamento de VPCs entre suas próprias Amazon VPCs ou com uma Amazon VPC em outra conta da AWS em uma única região. Para saber mais sobre o emparelhamento de Amazon VPCs, consulte a [documentação da VPC](#).

Note

A resolução de nomes DNS pode falhar para VPCs emparelhadas, dependendo das configurações aplicadas à VPC do ElastiCache. Para resolver isso, as duas VPCs devem ser habilitadas para nomes de hosts DNS e resolução DNS. Para obter mais informações, consulte [Habilitar a resolução de DNS para a conexão de emparelhamento da VPC](#).

Para acessar um cache em um Amazon VPC diferente por emparelhamento

1. Certifique-se de que as duas VPCs não tenham um intervalo de IP sobreposto, ou você não poderá compará-las.
2. Emparelhe as duas VPCs. Para obter mais informações, consulte [Criação e aceitação de uma conexão de emparelhamento da Amazon VPC](#).
3. Atualize sua tabela de roteamento. Para obter mais informações, consulte [Atualizar as tabelas de rotas para uma conexão de emparelhamento de VPC](#)

Veja a seguir a aparência das tabelas de rotas para o exemplo do diagrama anterior. Observe que pcx-a894f1c1 é a conexão de emparelhamento.

Destination	Target	Destination	Target
172.16.0.0/16	local	10.10.0.0/16	local
10.10.0.0/16	pcx-a894f1c1	0.0.0.0/0	igw-bfdcccd8
		172.16.0.0/16	pcx-a894f1c1

Tabela de roteamento de VPC

4. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada do grupo de segurança da aplicação na VPC emparelhada. Para obter mais informações, consulte a [Referência para security groups de VPC de emparelhamento](#).

O acesso a um cache por meio de uma conexão de emparelhamento implicará custos adicionais de transferência de dados.

Uso do Transit Gateway

Um Transit Gateway permite anexar VPCs e conexões VPN na mesma região da AWS e rotear tráfego entre elas. Um Transit Gateway funciona em contas da AWS, e você pode usar o Resource Access Manager da AWS para compartilhar o Transit Gateway com outras contas. Depois de compartilhar um gateway de trânsito com outra conta da AWS, o proprietário da conta poderá anexar as VPCs dele ao gateway de trânsito. Um usuário de qualquer uma das contas pode excluir o anexo a qualquer momento.

É possível ativar o multicast em um gateway de trânsito e, depois, criar um domínio de multicast do gateway de trânsito que permita ao tráfego de multicast ser enviado da origem de multicast para membros do grupo de multicast em anexos da VPC associados ao domínio.

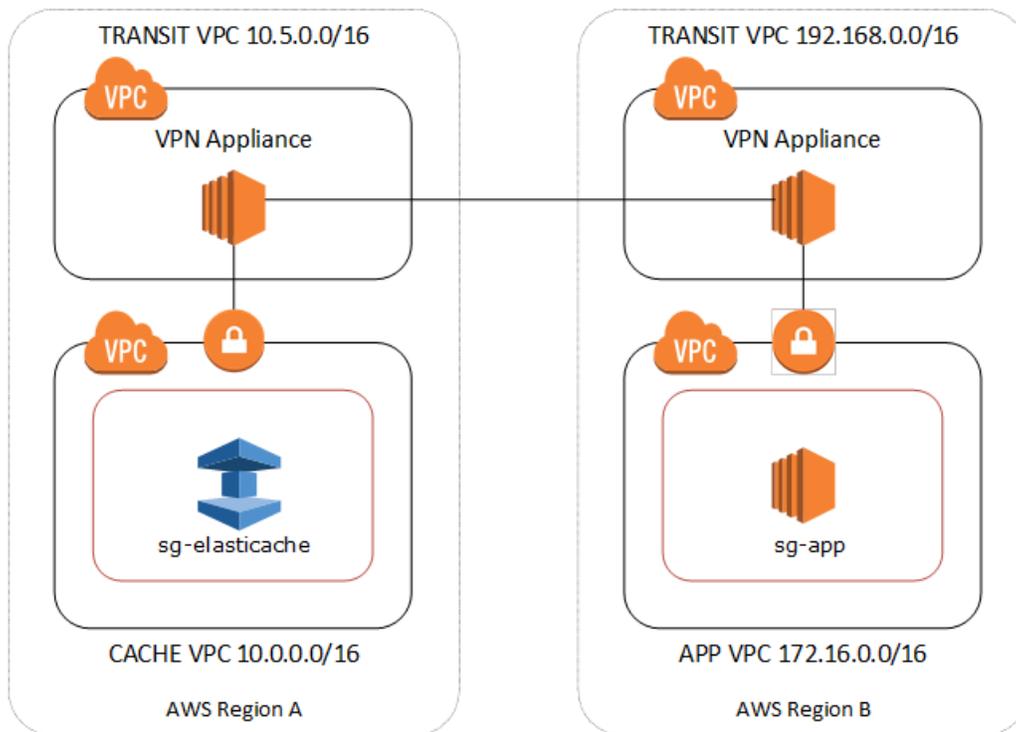
Também é possível criar um anexo da conexão de emparelhamento entre gateways de trânsito em diferentes regiões da AWS. Isso permite que você roteie o tráfego entre os anexos dos gateways de trânsito em regiões diferentes.

Para obter mais informações, consulte [Gateways de trânsito](#).

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs em regiões diferentes

Uso da VPC de trânsito

Uma alternativa ao uso do emparelhamento de VPC, outra estratégia comum para conectar várias VPCs geograficamente dispersas e redes remotas é criar uma VPC de trânsito que serve como um centro de trânsito de rede global. Uma VPC de trânsito simplifica o gerenciamento da rede e minimiza o número de conexões necessárias para conectar várias VPCs e redes remotas. Esse design pode economizar tempo e esforços e também reduzir custos, uma vez que é implementado praticamente sem as despesas tradicionais de estabelecer uma presença física em um hub de trânsito de colocação ou implantar equipamentos de rede física.



Conexão entre diferentes VPCs em regiões distintas

Depois de estabelecido o Amazon VPC de trânsito, uma aplicação implantada em uma VPC “spoke” em uma região pode se conectar a um cache do ElastiCache em uma VPC “spoke” em outra região.

Para acessar um cache em uma VPC diferente dentro de uma região da AWS diferente

1. Implante uma solução de VPC de trânsito. Para obter mais informações, consulte [Transit Gateway da AWS](#).
2. Atualize as tabelas de roteamento da VPC nas VPC de aplicativos e cache para rotear o tráfego através do VGW (gateway privado virtual) e do dispositivo de VPN. No caso do Roteamento dinâmico com o protocolo BGP, suas rotas podem ser propagadas automaticamente.
3. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada do intervalo IP de instâncias da aplicação. Observe que você não poderá fazer referência ao security group do servidor de aplicativos nesse cenário.

O acesso a um cache entre regiões introduzirá latências de rede e custos adicionais de transferência de dados entre regiões.

Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente

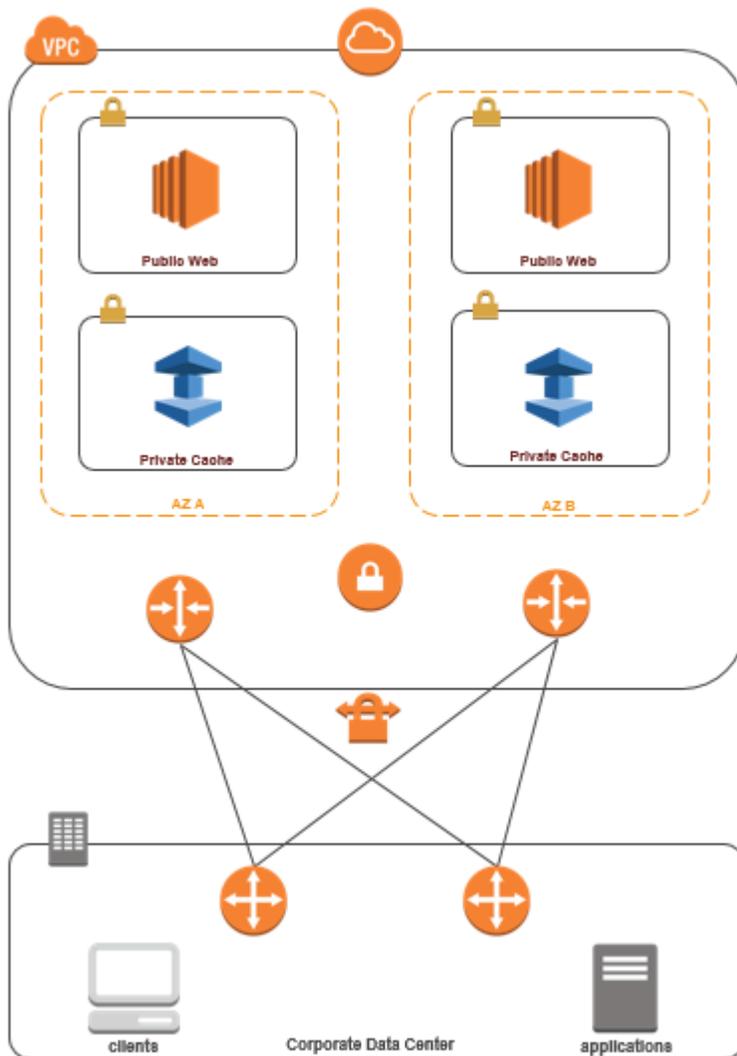
Outro cenário possível é uma arquitetura híbrida em que clientes ou aplicações no datacenter do cliente podem precisar acessar um cache do ElastiCache na VPC. Esse cenário também tem suporte, desde que haja conectividade entre a VPC dos clientes e o datacenter via VPN ou Direct Connect.

Tópicos

- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando conectividade de VPN](#)
- [Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando o Direct Connect](#)

Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando conectividade de VPN

O diagrama a seguir ilustra o acesso a um cache do ElastiCache a partir de uma aplicação executada na sua rede corporativa usando conexões VPN.



Conexão ao ElastiCache a partir do seu datacenter através de uma VPN

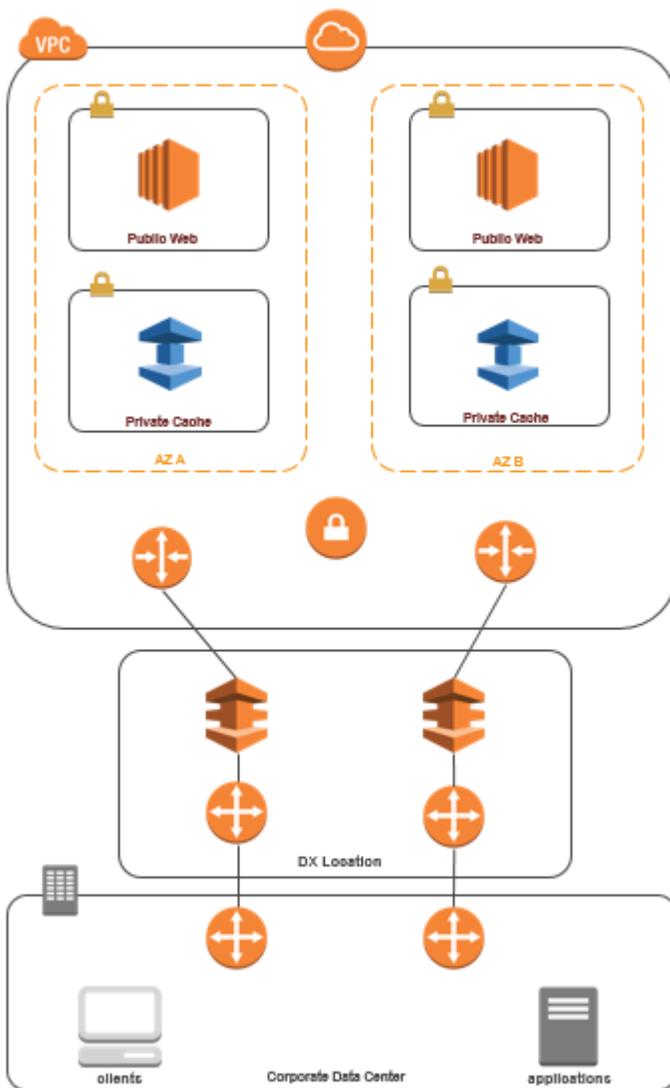
Para acessar um cache em uma VPC a partir do aplicativo no local via conexão VPN

1. Estabeleça a conectividade de VPN adicionando um gateway privado virtual de hardware à sua VPC. Para obter mais informações, consulte o tópico sobre como [Adicionar um gateway privado virtual de hardware à sua VPC](#).
2. Atualize a tabela de roteamento de VPC para a sub-rede na qual seu cache do ElastiCache está implantado para permitir o tráfego do seu servidor de aplicações on-premises. No caso do Roteamento dinâmico com o BGP, suas rotas podem ser propagadas automaticamente.
3. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada dos servidores de aplicações on-premises.

Acessar um cache através de uma conexão VPN introduzirá latências de rede e custos adicionais de transferência de dados.

Acesso a um cache do ElastiCache a partir de uma aplicação executada no datacenter de um cliente usando o Direct Connect

O diagrama a seguir ilustra o acesso a um cache do ElastiCache a partir de uma aplicação executada na sua rede corporativa usando o Direct Connect.



Conexão ao ElastiCache a partir do seu datacenter via Direct Connect

Para acessar um cache do ElastiCache de uma aplicação executada em sua rede usando o Direct Connect

1. Estabeleça a conectividade Direct Connect. Para obter mais informações, consulte [Conceitos básicos do Direct Connect da AWS](#).
2. Modifique o grupo de segurança do seu cache do ElastiCache para permitir a conexão de entrada dos servidores de aplicações on-premises.

O acesso a um cache por meio de uma conexão DX pode introduzir latências de rede e taxas adicionais de transferência de dados.

Criar uma nuvem privada virtual (VPC)

Neste exemplo, você cria uma Amazon VPC com uma sub-rede privada para cada zona de disponibilidade.

Criação de uma Amazon VPC (console)

1. Faça login no Console de Gerenciamento da AWS e abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc>.
2. No painel da VPC, escolha Criar VPC.
3. Em Recursos a serem criados, escolha VPC e mais.
4. Em Número de zonas de disponibilidade (ZAs), escolha o número de zonas de disponibilidade nas quais iniciar suas sub-redes.
5. Em Número de sub-redes públicas, escolha o número de sub-redes públicas que você deseja adicionar à sua VPC.
6. Em Número de sub-redes privadas, escolha o número de sub-redes públicas que você deseja adicionar à sua VPC.

Tip

Anote os identificadores das sub-redes e indique quais são públicas e quais são privadas. Você precisará dessas informações mais tarde quando ativar seus clusters de cache e adicionar uma instância do Amazon EC2 à sua Amazon VPC.

7. Crie um grupo de segurança da Amazon VPC. Você usará esse grupo para seu cluster de cache e sua instância do Amazon EC2.
 - a. No painel de navegação do console de gerenciamento da Amazon VPC, escolha Security Groups (Grupos de segurança).
 - b. Escolha Criar grupo de segurança.
 - c. Digite um nome e uma descrição do seu security group nas caixas correspondentes. Na caixa VPC, escolha o identificador da sua Amazon VPC.

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](#)
my-vpc-security-group
Name cannot be edited after creation.

Description [Info](#)
my vpc security group

VPC [Info](#)
vpc-862574fc

Inbound rules [Info](#)

This security group has no inbound rules.

[Add rule](#)

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic	All	All	Custom 0.0.0.0/0	

[Add rule](#) [Delete](#)

- 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 instância do Amazon EC2 usando Secure Shell (SSH).
 - a. Na lista de navegação, escolha Security Groups.
 - b. Localize seu security group na lista e escolha-o.
 - c. Em Security Group, escolha a guia Inbound. Na caixa Create a new rule, escolha SSH e depois Add Rule.
 - d. Defina os seguintes valores para a sua nova regra de entrada a fim de permitir o acesso HTTP.
 - Tipo: HTTP
 - Origem: 0.0.0.0/0

Escolha Apply Rule Changes.

Agora, você está pronto para criar um grupo de sub-redes de cache e executar um cluster de cache na sua Amazon VPC.

- [Criação de um grupo de sub-redes](#)
- [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#).

Conexão a um cache em execução em um Amazon VPC

Este exemplo mostra como iniciar uma instância do Amazon EC2 na sua Amazon VPC. Você pode então fazer login nessa instância e acessar o cache do ElastiCache que está sendo executado no Amazon VPC.

Conexão a um cache em execução em um Amazon VPC (console)

Neste exemplo, crie uma instância do Amazon EC2 em sua Amazon VPC. Você pode usar essa instância do Amazon EC2 para se conectar a nós de cache em execução na Amazon VPC.

Note

Para obter informações sobre como usar o Amazon EC2, consulte o [Guia de conceitos básicos do Amazon EC2](#) na [Documentação do Amazon EC2](#).

Para criar uma instância do Amazon EC2 em sua Amazon VPC usando o console do Amazon EC2

1. Faça login no AWS Management Console e abra o console Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. No console, escolha Executar instância e siga estas etapas:
3. Na página Selecione uma Imagem de máquina da Amazon (AMI), escolha a AMI Amazon Linux de 64 bits e Selecionar.
4. Na página Choose an Instance Type (Escolher um tipo de instância), escolha 3. Configure Instance (Configurar instância).
5. Na página Configure os detalhes da instância, faça as seguintes seleções:
 - a. Na lista Network (Rede), escolha sua Amazon VPC.
 - b. Na lista Sub-rede, escolha sua sub-rede pública.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances ⓘ

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 (Marcar instância), digite um nome para sua instância do Amazon EC2 e depois 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 de segurança do Amazon EC2 para instâncias do 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 segurança da Amazon VPC depois escolha Review and Launch (Analisar e iniciar).

9. Na página Revisar instância e executar, escolha Executar.
10. Na janela Select an existing key pair or create a new key pair, especifique um par de chaves que você deseja usar com essa instância.

Note

Para obter mais informações sobre o gerenciamento de pares de chaves, consulte o [Guia de conceitos básicos do Amazon EC2](#).

11. Quando estiver pronto para executar sua instância do Amazon EC2, escolha Launch (Iniciar).

Agora, você pode atribuir um endereço IP elástico à instância do Amazon EC2 que acabou de criar. Você precisa usar esse endereço IP para se conectar à instância do Amazon EC2.

Como atribuir um endereço IP elástico (console)

1. Abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.
2. Na lista de navegação, escolha IPs elásticos.
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 (Associar endereço), na caixa Instance (Instância), escolha o ID da instância do Amazon EC2 que você iniciou.

Na caixa Endereço IP privado, selecione a caixa para obter o endereço IP privado e escolha Associar.

Agora, você pode usar o SSH para se conectar à instância do Amazon EC2 usando o endereço IP elástico criado.

Para se conectar à sua instância do Amazon EC2

- 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 faça logout da instância do Amazon EC2 ainda.

Agora, você está pronto para interagir com o seu cluster do ElastiCache. 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)

1. 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!
```

2. Use o telnet para se conectar ao endpoint do nó de cache pela porta 6379. Substitua o nome do host mostrado abaixo pelo nome do host do nó de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Agora você está conectado ao mecanismo de cache e pode emitir comandos. Neste exemplo, você adiciona um item de dados ao cache e o obtém logo em seguida. Por último, você se desconectará do nó do cache.

Para armazenar uma chave e um valor, digite as duas linhas a seguir:

```
set mykey myvaLue
```

O mecanismo de cache responde com o seguinte:

```
OK
```

Para recuperar o valor para mykey, digite o seguinte:

```
get mykey
```

Para desconectar-se do mecanismo de cache, digite o seguinte:

```
quit
```

3. Acesse o console do ElastiCache em <https://console.aws.amazon.com/elasticache/> e obtenha o endpoint para um dos nós do seu cluster de cache. Para obter mais informações, consulte [Localização de endpoints de conexão](#) no Redis.
4. Use o telnet para se conectar ao endpoint do nó de cache pela porta 6379. Substitua o nome do host mostrado abaixo pelo nome do host do nó de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Agora você está conectado ao mecanismo de cache e pode emitir comandos. Neste exemplo, você adiciona um item de dados ao cache e o obtém logo em seguida. Por último, você se desconectará do nó do cache.

Para armazenar uma chave e um valor, digite o seguinte:

```
set mykey myvaLue
```

O mecanismo de cache responde com o seguinte:

```
OK
```

Para recuperar o valor para mykey, digite o seguinte:

```
get mykey
```

O mecanismo de cache responde com o seguinte:

```
get mykey  
myvalue
```

Para desconectar-se do mecanismo de cache, digite o seguinte:

```
quit
```

Important

Para evitar cobranças adicionais na sua conta da AWS, certifique-se de excluir todos os recursos da AWS que você não deseja mais depois de tentar esses exemplos.

API do Amazon ElastiCache e endpoints da VPC de interface (AWS PrivateLink)

É possível estabelecer uma conexão privada entre a VPC e os endpoints da API do Amazon ElastiCache criando um endpoint da VPC de interface. Os endpoints de interface são desenvolvidos pelo [AWS PrivateLink](#). O AWS PrivateLink permite que você acesse de forma privada as operações da API do Amazon ElastiCache sem um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect.

As instâncias na sua VPC não precisam de endereços IP públicos para a comunicação com os endpoints da API do Amazon ElastiCache. As instâncias também não precisam de endereços IP públicos para usar qualquer uma das operações da API do ElastiCache disponíveis. O tráfego entre seu VPC e Amazon ElastiCache não sai da rede da 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 do usuário do Amazon EC2.

- Para obter mais informações sobre limites de VPC , consulte [VPC endpoints de interface \(AWS PrivateLink\)](#) no Guia do usuário da Amazon VPC.
- Para obter mais informações sobre como chamar operações da API do ElastiCache, consulte [Operações da API do ElastiCache](#).

Depois de criar um endpoint da VPC de interface, se você habilitar nomes de host [DNS privados](#) para o endpoint, o endpoint padrão do ElastiCache (<https://elasticache.Region.amazonaws.com>) será resolvido para o endpoint da VPC. Se você não habilitar nomes de host DNS privados, o Amazon VPC fornecerá um nome de endpoint DNS que poderá ser usado no seguinte formato:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Para obter mais informações, consulte [Endpoints da VPC de interface \(AWS PrivateLink\)](#) no Manual do usuário do Amazon VPC. O ElastiCache oferece suporte a chamadas para todas as suas [ações de API](#) dentro da VPC.

Note

Os nomes de host DNS privados podem ser habilitados para apenas um endpoint da VPC na VPC. Se você quiser criar um endpoint da VPC adicional, o nome de host DNS privado deve ser desabilitado para ele.

Considerações sobre endpoints da VPC do

Antes de configurar um endpoint da VPC de interface para endpoints da API do Amazon ElastiCache, revise [Propriedades e limitações do endpoint de interface](#) no Guia do usuário da Amazon VPC. Todas as operações da API do ElastiCache relevantes para o gerenciamento de recursos do Amazon ElastiCache estão disponíveis na VPC usando AWS PrivateLink.

As políticas de endpoint da VPC têm suporte para endpoints da API do ElastiCache. Por padrão, o acesso total às operações de API do ElastiCache é permitido por meio do endpoint. Para obter mais informações, consulte [Controlar o acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

Criar um endpoint da VPC de interface para a API do ElastiCache

É possível criar um endpoint da VPC para o serviço de APIs do Amazon ElastiCache usando o console da Amazon VPC ou a AWS CLI. Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário da Amazon VPC.

Depois de criar um endpoint da VPC de interface, será possível habilitar nomes de host DNS privados para o endpoint. Ao fazer isso, o endpoint padrão do Amazon ElastiCache (<https://elasticache.Região.amazonaws.com>) resolve para seu endpoint da VPC. Para as regiões da AWS China (Pequim) e China (Ningxia), é possível fazer solicitações de API com o endpoint da VPC usando elasticache.cn-north-1.amazonaws.com.cn para Pequim e elasticache.cn-northwest-1.amazonaws.com.cn para Ningxia. Para obter mais informações, consulte [Acessar um serviço por um endpoint de interface](#) no Guia do usuário da Amazon VPC.

Criar uma política de endpoint da VPC para a API do Amazon ElastiCache

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso à API do ElastiCache. 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 [Controlar o acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

Exemplo Política de endpoint da VPC para ações de API do ElastiCache

Veja a seguir um exemplo de uma política de endpoint da API do ElastiCache. Quando anexada a um endpoint, essa política concede acesso às ações indicadas da API do ElastiCache para todos os principais em todos os recursos.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster",
```

```
"elasticache:CreateSnapshot"
],
"Resource": "*"
}]
}
```

Exemplo Política de endpoint da VPC que nega todo o acesso de uma conta da AWS especificada

A política de VPC endpoint a seguir nega à conta da AWS **123456789012** todos os acessos aos recursos que usam o endpoint. A política permite todas as ações de outras contas.

```
{
  "Statement": [{
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*",
    "Principal": "*"
  },
  {
    "Action": "*",
    "Effect": "Deny",
    "Resource": "*",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    }
  }
]
}
```

Sub-redes e grupos de sub-redes

Um grupo de sub-redes é um conjunto de sub-redes (normalmente privadas) que você pode designar para seus clusters autoprojeto em execução em um ambiente Amazon Virtual Private Cloud (VPC).

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

O ElastiCache fornece um grupo de sub-redes IPv4 padrão ou é possível optar por criar um novo. Para o IPv6, é necessário criar um grupo de sub-rede com um bloco CIDR do IPv6. Se você escolher dual stack (pilha dupla), deverá selecionar um tipo de IP de descoberta: IPv6 ou IPv4.

O ElastiCache Sem Servidor não usa um recurso do grupo de sub-redes e, em vez disso, obtém uma lista de sub-redes diretamente durante a criação.

Esta seção discute como criar e utilizar sub-redes e grupos de sub-rede para gerenciar o acesso aos seus recursos do ElastiCache.

Para obter mais informações sobre o uso de grupos de sub-redes em um ambiente da Amazon VPC, consulte [Acesso ao seu cluster ou grupo de replicação](#).

Tópicos

- [Criação de um grupo de sub-redes](#)
- [Atribuição de um grupo de sub-redes a um cache](#)
- [Modificação de um grupo de sub-redes](#)
- [Exclusão de um grupo de sub-redes](#)

Criação de um grupo de sub-redes

Um grupo de sub-redes de cache é uma coleção de sub-redes que você pode querer designar aos seus caches em uma VPC. Ao executar um cache em uma VPC, é necessário selecionar um grupo de sub-redes de cache. Em seguida, o ElastiCache usa esse grupo de sub-rede de cache para atribuir endereços IP dentro dessa sub-rede a cada nó de cache do cache.

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

Se você escolher IPV4 como seu tipo de redes, um grupo de sub-rede padrão estará disponível ou é possível optar por criar um novo. O ElastiCache usa esse grupo de sub-redes para escolher uma sub-rede e endereços IP dentro dessa sub-rede para associar aos seus nós. Se você escolher pilha dupla ou IPV6, será direcionado para criar sub-redes de pilha dupla ou IPV6. Para obter mais informações sobre os tipos de rede, consulte [Network type](#). Para obter mais informações, consulte [Criar uma sub-rede na sua VPC](#).

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

Criação de um grupo de sub-redes (console)

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

Como criar um grupo de sub-redes (console)

1. Faça login no Console de Gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Na lista de navegação, escolha Grupos de sub-redes.
3. Selecione Create subnet group (Criar grupo de sub-redes).
4. No assistente Criar grupo de sub-redes, faça o seguinte. Quando estiver satisfeito com as configurações, escolha Criar.
 - a. Na caixa Name (Nome), digite um nome para o seu grupo de sub-redes.
 - b. Na caixa Description (Descrição), digite uma descrição para seu grupo de sub-redes.

- c. Na caixa ID da VPC, escolha o Amazon VPC.
 - d. Todas as sub-redes são escolhidas por padrão. No painel Sub-redes selecionadas, clique em Gerenciar e selecione as zonas de disponibilidade ou as [Zonas locais](#) e os IDs das sub-redes privadas e selecione Escolher.
5. Na mensagem de confirmação exibida, escolha Close (Fechar).

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

Criação de um grupo de sub-redes (AWS CLI)

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

Para Linux, macOS ou Unix:

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

Para Windows:

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

Esse comando deve produzir um resultado semelhante ao seguinte:

```
{  
  "CacheSubnetGroup": {  
    "VpcId": "vpc-37c3cd17",  
    "CacheSubnetGroupDescription": "Testing",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-53df9c3a",  
        "SubnetAvailabilityZone": {
```

```
        "Name": "us-west-2a"
      }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
  }
}
```

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

Atribuição de um grupo de sub-redes a um cache

Depois de criar um grupo de sub-redes, você pode iniciar um cache em um Amazon VPC. Para obter mais informações, consulte.

- Cluster Redis autônomo: para ativar um cluster do Redis de nó único, consulte [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#). Na etapa 7.a (Configurações avançadas do Redis), escolha um grupo de sub-redes da VPC.
- Grupo de replicação do Redis (modo cluster desabilitado): para ativar um grupo de replicação do Redis (modo cluster desabilitado) em uma VPC, consulte [Criação de um grupo de replicação do Redis \(modo cluster desabilitado\) do princípio](#). Na etapa 7.b (Configurações avançadas do Redis), escolha um grupo de sub-redes da VPC.
- Grupo de replicação do Redis (modo cluster habilitado): [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#). Na etapa 6.i (Configurações avançadas do Redis), escolha um grupo de sub-redes da VPC.

Modificação de um grupo de sub-redes

Você pode modificar a descrição de um grupo de sub-redes ou modificar a lista de IDs de sub-rede associados ao grupo de sub-redes. Você não poderá excluir um ID de sub-rede de um grupo de sub-redes se um cache estiver usando essa sub-rede atualmente.

Os procedimentos a seguir mostram como modificar um grupo de sub-redes.

Modificação de grupos de sub-redes (console)

Para modificar um grupo de sub-redes

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Grupos de sub-redes.
3. Na lista de grupos de sub-redes, selecione o botão de opção daquele que você deseja modificar e escolha Modificar.
4. No painel Sub-redes selecionadas, escolha Gerenciar.
5. Faça qualquer alteração nas sub-redes selecionadas e clique em Escolher.
6. Clique em Salvar alterações para salvá-las.

Modificação de grupos de sub-redes (AWS CLI)

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

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup \  
  --cache-subnet-group-description "New description" \  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Para Windows:

```
aws elasticache modify-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "New description" ^  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Esse comando deve produzir um resultado semelhante ao seguinte:

```
{
  "CacheSubnetGroup": {
    "VpcId": "vpc-73cd3c17",
    "CacheSubnetGroupDescription": "New description",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-42dcf93a",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      },
      {
        "SubnetIdentifier": "subnet-48fc12a9",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
  }
}
```

Para obter mais informações, consulte o tópico sobre a AWS CLI [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 console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Grupos de sub-redes.
3. Na lista de grupos de sub-redes, escolha aquele que deseja excluir e selecione Delete.
4. Quando você precisar confirmar essa operação, digite o nome do grupo de sub-redes no campo de entrada de texto e escolha Excluir.

Exclusão de um grupo de sub-redes (AWS CLI)

Usando o AWS CLI, chame o comando `delete-cache-subnet-group` com o seguinte parâmetro:

- `--cache-subnet-group-name` *mysubnetgroup*

Para Linux, macOS ou Unix:

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup
```

Para Windows:

```
aws elasticache delete-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup
```

Este comando não produz saída.

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

Gerenciamento de identidade e acesso no Amazon ElastiCache

O AWS Identity and Access Management (IAM) é um AWS service (Serviço da AWS) que ajuda o administrador a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) para usar os recursos do ElastiCache. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Como gerenciar acesso usando políticas](#)
- [Como o Amazon ElastiCache funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para o Amazon ElastiCache](#)
- [Solução de problemas de identidade e acesso do Amazon ElastiCache](#)
- [Controle de acesso](#)
- [Visão geral do gerenciamento de permissões de acesso aos recursos do ElastiCache](#)

Público

O uso do AWS Identity and Access Management (IAM) varia dependendo do trabalho que for realizado no ElastiCache.

Usuário do serviço: se você usar o serviço ElastiCache para fazer o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que usar mais recursos do ElastiCache para fazer seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no ElastiCache, consulte [Solução de problemas de identidade e acesso do Amazon ElastiCache](#).

Administrador do serviço: se você for o responsável pelos recursos do ElastiCache na empresa, provavelmente terá acesso total ao ElastiCache. Cabe a você determinar quais funcionalidades e recursos do ElastiCache os usuários do serviço devem acessar. Assim, você deve enviar solicitações

ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como a empresa pode usar o IAM com o ElastiCache, consulte [Como o Amazon ElastiCache funciona com o IAM](#).

Administrador do IAM: se você é um administrador do IAM, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao ElastiCache. Para visualizar exemplos de políticas baseadas em identidade do ElastiCache que podem ser usadas no IAM, consulte [Exemplos de políticas baseadas em identidade para o Amazon ElastiCache](#).

Autenticando com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. É necessário ser autenticado (fazer login na AWS) como o usuário raiz da Usuário raiz da conta da AWS, como usuário do IAM ou assumindo um perfil do IAM.

Você pode fazer login na AWS como uma identidade federada usando credenciais fornecidas por uma fonte de identidades. Os usuários do AWS IAM Identity Center (IAM Identity Center), a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades utilizando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

É possível fazer login no ou no portal de acesso da AWS Management Console dependendo do tipo de usuário que você é. Para obter mais informações sobre como fazer login na AWS, consulte [Como fazer login na conta da Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS.

Se você acessar a AWS programaticamente, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface da linha de comando (CLI) para você assinar criptograficamente as solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte [Assinar solicitações de API da AWS](#) no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça mais informações de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do usuário do AWS IAM Identity Center e [Usar a autenticação multifator \(MFA\) na AWS](#) no Guia do usuário do IAM.

Usuário raiz da Conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos os atributos e Serviços da AWS na conta. Essa identidade, denominada usuário raiz da Conta da AWS, e é acessada por login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não utilizar 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 obter a lista completa das tarefas que exigem login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários, inclusive os que precisam de acesso de administrador, usem a federação com um provedor de identidades para acessar Serviços da AWS usando credenciais temporárias.

Identidade federada é um usuário de seu diretório de usuários corporativos, um provedor de identidades da web AWS Directory Service, o , o diretório do Centro de Identidade ou qualquer usuário que acesse os Serviços da AWS usando credenciais fornecidas por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem perfis que fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o .AWS IAM Identity Center. Você pode criar usuários e grupos no Centro de Identidade do IAM ou se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todas as suas Contas da AWS e aplicações. Para obter mais informações sobre o Centro de Identidade do IAM, consulte [“O que é o Centro de Identidade do IAM?”](#) no Guia do usuário do AWS IAM Identity Center.

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicação. Sempre que possível, recomendamos contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de utilização específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exijam credenciais](#) de longo prazo no Guia do usuário do IAM.

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível utilizar 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 IAMAdmins e atribuir a esse grupo permissões para administrar atributos do IAM.

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 usuário do IAM \(em vez de uma função\)](#) no Guia do usuário do IAM.

Perfis do IAM

Um [perfil do IAM](#) é uma identidade dentro da Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. É possível assumir temporariamente um perfil do IAM no AWS Management Console [alternando perfis](#). É possível assumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre métodos para o uso de perfis, consulte [Usar perfis do IAM](#) no Guia do usuário do IAM.

Perfis do IAM 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 mais informações sobre perfis para federação, consulte [Criar uma função para um provedor de identidade de terceiros](#) no Guia do usuário do IAM. Se você usar o IAM Identity Center, deverá configurar um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de permissões](#) no Guia do usuário do AWS IAM Identity Center.
- **Permissões temporárias para usuários do IAM:** um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas:** é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem que você anexe uma política diretamente a um atributo (em vez de usar um perfil como proxy). Para saber a diferença entre perfis e políticas baseadas em atributo para acesso entre contas, consulte [Como os perfis do IAM diferem das políticas baseadas em atributo](#) no Guia do usuário do IAM.

- **Acesso entre serviços:** alguns Serviços da AWS usam atributos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no 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 ao serviço.
- **Encaminhamento de sessões de acesso (FAS):** qualquer pessoa que utilizar uma função ou usuário do IAM para realizar ações na AWS é considerada uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).
- **Perfil de serviço:** um perfil de serviço é um [perfil do IAM](#) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.
- **Função vinculada ao serviço:** uma função vinculada a serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir o perfil para executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode exibir, mas não pode editar as permissões para perfis vinculados ao serviço.
- **Aplicações em execução no Amazon EC2:** é possível usar um perfil do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da AWS API. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir um perfil da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém a perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte [Usar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2](#) no Guia do usuário do IAM.

Para saber se deseja usar os perfis do IAM, consulte [Quando criar um perfil do IAM \(em vez de um usuário\)](#) no Guia do usuário do IAM.

Como gerenciar acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as a identidades ou atributos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou atributo, define suas permissões. A AWS avalia essas políticas quando uma entidade principal (usuário, usuário raiz ou sessão de perfil) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas são armazenadas na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Os administradores podem usar AWS as políticas JSON da para especificar quem tem acesso a 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 aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independente do método usado para executar 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 perfil do AWS Management Console, da AWS CLI ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. 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 [Criar políticas do IAM](#) no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas embutidas ou políticas gerenciadas. As políticas embutidas são anexadas diretamente a um único usuário, grupo ou função. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e perfis na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recurso

Políticas baseadas em atributos são documentos de políticas JSON que você anexa a um atributo. São exemplos de políticas baseadas em recurso as políticas de confiança de função do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recurso, 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 uma entidade principal especificado pode executar nesse recurso e em que condições. Você precisa [especificar uma entidade principal](#) em uma política baseada em recurso. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Políticas baseadas em atributos são políticas em linha que estão localizadas nesse serviço. Não é possível usar as políticas gerenciadas da AWS do IAM em uma política baseada em atributos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recurso, embora não usem o formato de documento de política JSON.

Amazon S3, AWS WAF e Amazon VPC são exemplos de serviços compatíveis com ACLs. Para saber mais sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS aceita tipos de política 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 atributo avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). 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 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 identidades do IAM](#) no Guia do usuário do IAM.

- Políticas de controle de serviço (SCPs): SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS pertencentes à sua empresa. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, incluindo cada Usuário raiz da conta da AWS. Para obter mais informações sobre as Organizações e SCPs, consulte [How SCPs work \(Como os SCPs funcionam\)](#) no AWS Organizations Guia do usuário do .
- 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 do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir uma solicitação quando há vários tipos de política envolvidos, consulte [Lógica da avaliação](#) de políticas no Guia do usuário do IAM.

Como o Amazon ElastiCache funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao ElastiCache, saiba quais recursos do IAM estão disponíveis para uso com o ElastiCache.

Recursos do IAM que você pode usar com o Amazon ElastiCache

atributo do IAM	Suporte ao ElastiCache
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim

atributo do IAM	Suporte ao ElastiCache
atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Sim
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados ao serviço	Sim

Para obter uma visualização de alto nível de como o ElastiCache e outros serviços da AWS funcionam com a maioria dos recursos do IAM, consulte [Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade para o ElastiCache

Suporta políticas baseadas em identidade	Sim
--	-----

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos 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 função à qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elementos da política JSON do IAM](#) no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o ElastiCache

Para visualizar exemplos de políticas baseadas em identidade do ElastiCache, consulte [Exemplos de políticas baseadas em identidade para o Amazon ElastiCache](#).

Políticas baseadas em recursos no ElastiCache

Oferece suporte a políticas baseadas em recurso	Não
---	-----

Políticas baseadas em atributos são documentos de políticas JSON que você anexa a um atributo. São exemplos de políticas baseadas em recurso as políticas de confiança de função do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recurso, 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 uma entidade principal especificado pode executar nesse recurso e em que condições. Você precisa [especificar uma entidade principal](#) em uma política baseada em recurso. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando a entidade principal e o atributo estão em diferentes Contas da AWS, um administrador do IAM da conta confiável também deve conceder à entidade principal (usuário ou perfil) permissão para acessar o atributo. 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 outra política baseada em identidade será necessária. Para obter mais informações, consulte [Como as funções do IAM diferem de políticas baseadas em recurso](#) no Guia do usuário do IAM.

Ações de políticas para o ElastiCache

Oferece suporte a ações de políticas	Sim
--------------------------------------	-----

Os administradores podem usar as políticas de JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. 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 das ações do ElastiCache, consulte [Actions Defined by Amazon ElastiCache](#) na Referência de autorização do serviço.

As ações de políticas no 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 visualizar exemplos de políticas baseadas em identidade do ElastiCache, consulte [Exemplos de políticas baseadas em identidade para o Amazon ElastiCache](#).

Recursos de políticas para o ElastiCache

Oferece suporte a atributos de políticas

Sim

Os administradores podem usar as políticas de JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política Resource JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento Resource ou um elemento NotResource. Como prática recomendada, especifique um atributo usando seu [Nome do atributo da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um caractere curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos do ElastiCache e seus ARNs, consulte [Resources Defined by Amazon ElastiCache](#) na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Actions Defined by Amazon ElastiCache](#).

Para visualizar exemplos de políticas baseadas em identidade do ElastiCache, consulte [Exemplos de políticas baseadas em identidade para o Amazon ElastiCache](#).

Chaves de condição de políticas para o ElastiCache

Suporta chaves de condição de política específicas de serviço	Sim
---	-----

Os administradores podem usar as políticas de JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento Condition (ou Condition bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento Condition é opcional. É possível criar expressões condicionais que usam [atendentes 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, a AWS avaliará a condição usando uma operação lógica OR. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode utilizar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos de política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

A AWS oferece suporte a chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as AWS chaves de condição globais da , consulte [AWSChaves de contexto de condição globais da](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do ElastiCache, consulte [Condition Keys for Amazon ElastiCache](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar a chave de condição, consulte [Ações definidas pelo Amazon ElastiCache](#).

Para visualizar exemplos de políticas baseadas em identidade do ElastiCache, consulte [Exemplos de políticas baseadas em identidade para o Amazon ElastiCache](#).

Listas de controle de acesso (ACLs) no ElastiCache

Oferece suporte a ACLs	Sim
------------------------	-----

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recurso, embora não usem o formato de documento de política JSON.

Controle de acesso por atributo (ABAC) com o ElastiCache

Oferece suporte a ABAC (tags em políticas)	Sim
--	-----

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou perfis) e a muitos atributos da AWS. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações onde o gerenciamento de políticas se torna um problema.

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 `aws:TagKeys` chaves de condição.

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 o ABAC, consulte [O que é ABAC?](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\)](#) no Guia do usuário do IAM.

Usar credenciais temporárias com o ElastiCache

Oferece suporte a 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 que funcionam com o IAM](#) no Guia do usuário do IAM.

Você está usando credenciais temporárias se faz login no AWS Management Console usando qualquer método, exceto um nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (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 como alternar perfis, consulte [Alternar para um perfil \(console\)](#) no Guia do usuário do IAM.

Você pode criar credenciais temporárias manualmente usando a AWS CLI ou a API da AWS. Em seguida, você pode usar essas credenciais temporárias para acessar a AWS. A 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 no IAM](#).

Permissões de entidade principal entre serviços para o ElastiCache

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim
--	-----

Quando você usa um usuário ou perfil do IAM para executar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).

Perfis de serviço para o ElastiCache

Oferece suporte a perfis de serviço	Sim
-------------------------------------	-----

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

Warning

Alterar as permissões de um perfil de serviço pode interromper a funcionalidade do ElastiCache. Edite perfis de serviço somente quando o ElastiCache fornecer orientação para isso.

Funções vinculadas ao serviço criadas para o ElastiCache

Oferece suporte a perfis vinculados ao serviço	Sim
--	-----

Um perfil vinculado ao serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode presumir a função de executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte [Serviços do AWS que funcionam com o 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 esse serviço.

Exemplos de políticas baseadas em identidade para o Amazon ElastiCache

Por padrão, os usuários e os perfis não têm permissões para criar ou modificar os recursos do ElastiCache. Eles também não podem executar tarefas usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API AWS. Para conceder aos usuários permissão para executar ações nos recursos de que precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte [Criação de políticas do IAM](#) no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recurso definidos pelo ElastiCache, por exemplo, o formato dos ARNs para cada um dos tipos de recurso, consulte [Ações, recursos e chaves de condição do Amazon ElastiCache](#) na Referência de autorização do serviço.

Tópicos

- [Melhores práticas de políticas](#)
- [Uso do console do ElastiCache](#)
- [Permitir que usuários visualizem suas próprias permissões](#)

Melhores práticas de políticas

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do ElastiCache em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com AWS as políticas gerenciadas pela e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as AWS políticas gerenciadas pela que concedem permissões para muitos casos de uso comuns. Elas estão disponíveis na sua Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente da AWS específicas para seus casos de uso. Para

obter mais informações, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para perfis de trabalho](#) no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e atributos. 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 a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte [Elementos de política JSON do IAM: Condition](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de 100 verificações de política e recomendações acionáveis para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário do IAM.
- Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Uso do console do ElastiCache

Para acessar o console do Amazon ElastiCache, você deve ter um conjunto mínimo de permissões. Essas permissões concedem autorização para que você liste e visualize detalhes sobre os recursos do ElastiCache na sua 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.

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Para garantir que os usuários e os perfis ainda possam usar o console do ElastiCache, anexe também a política `ConsoleAccess` ou `ReadOnly` do ElastiCache gerenciada pela AWS às entidades. Para obter mais informações, consulte [Adicionando Permissões a um Usuário](#) no Guia do Usuário do IAM.

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como é possível criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a 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 identidade e acesso do Amazon ElastiCache

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o ElastiCache e o IAM.

Tópicos

- [Não tenho autorização para executar uma ação no ElastiCache](#)
- [Não estou autorizado a executar iam:PassRole](#)
- [Desejo permitir que pessoas fora da minha conta da AWS acessem meus recursos do ElastiCache](#)

Não tenho autorização para executar uma ação no ElastiCache

Se o AWS Management Console informar que você não está autorizado a executar uma ação, você deverá entrar em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você 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 para permitir a ele o acesso ao recurso *my-example-widget* usando a ação `elasticache:GetWidget`.

Não estou autorizado a executar iam:PassRole

Se você receber um erro informando que não tem autorização para executar a ação `iam:PassRole`, as políticas deverão ser atualizadas para permitir que você passe um perfil para o ElastiCache.

Alguns Serviços da AWS permitem que você transmita um perfil existente para o serviço, em vez de criar um perfil de serviço ou um perfil vinculado ao serviço. Para fazer isso, um usuário deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no ElastiCache. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar a função para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu a você suas credenciais de login.

Desejo permitir que pessoas fora da minha conta da AWS acessem meus recursos do ElastiCache

Você pode criar uma função 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 a função. 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 o seguinte:

- Para saber se o ElastiCache oferece suporte a esses recursos, consulte [Como o Amazon ElastiCache funciona com o IAM](#).
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte [Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você](#) no Guia de usuário do IAM.
- Para saber como conceder acesso a seus recursos para terceiros Contas da AWS, consulte [Fornecimento de acesso a Contas da AWS pertencentes a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.

- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

Controle de acesso

Você pode ter credenciais válidas para autenticar suas solicitações, mas a menos que tenha permissões, você não pode criar ou acessar recursos do ElastiCache. Por exemplo, você deve ter permissões para criar um cluster do ElastiCache.

As seções a seguir descrevem como gerenciar permissões para o ElastiCache. Recomendamos que você leia a visão geral primeiro.

- [Visão geral do gerenciamento de permissões de acesso aos recursos do ElastiCache](#)
- [Uso de políticas baseadas em identidade \(políticas do IAM\) para o Amazon ElastiCache](#)

Visão geral do gerenciamento de permissões de acesso aos recursos do ElastiCache

Cada recurso da AWS pertence a uma conta da AWS, e as permissões para criá-lo ou acessá-lo são regidas por políticas de permissões. Um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções). Além disso, o Amazon ElastiCache também oferece suporte à anexação de políticas de permissões aos recursos.

Note

Um administrador da conta (ou usuário administrador) é um usuário com privilégios de administrador. Para obter mais informações, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

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

- Usuários e grupos no AWS IAM Identity Center:

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

- Usuários gerenciados no IAM com provedor de identidades:

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

- Usuários do IAM:

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

Tópicos

- [Recursos e operações do Amazon ElastiCache](#)
- [Informações sobre propriedade de recursos](#)
- [Gerenciamento de acesso aos recursos](#)

- [Políticas gerenciadas pela AWS para o Amazon ElastiCache](#)
- [Uso de políticas baseadas em identidade \(políticas do IAM\) para o Amazon ElastiCache](#)
- [Permissões em nível de recurso](#)
- [Uso de chaves de condição](#)
- [Uso de funções vinculadas ao serviço para o Amazon ElastiCache](#)
- [Permissões da API do ElastiCache: referência de ações, recursos e condições](#)

Recursos e operações do Amazon ElastiCache

Para ver uma lista dos tipos de recursos do ElastiCache e seus ARNs, consulte [Resources Defined by Amazon ElastiCache](#) na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Actions Defined by Amazon ElastiCache](#).

Informações sobre propriedade de recursos

Um proprietário do recurso é a conta da AWS que criou o recurso. Ou seja, o proprietário do recurso é a conta da AWS da entidade principal que autentica a solicitação que cria o recurso. Uma entidade principal pode ser a conta raiz, um usuário do IAM ou uma função do IAM. Os seguintes exemplos mostram como isso funciona:

- Suponha que você use as credenciais da conta raiz da sua conta da AWS para criar um cluster de cache. Nesse caso, sua conta da AWS é o proprietário do recurso. No ElastiCache, o recurso é o cluster de cache.
- Suponha que você crie um usuário do IAM na sua conta da AWS e conceda permissões para criar um cluster de cache para ele. Nesse caso, o usuário pode criar um cluster de cache. No entanto, sua conta da AWS, à qual o usuário pertence, é a proprietária do recurso de cluster de cache.
- Suponha que você crie uma função do IAM na sua conta da AWS com permissões para criar um cluster de cache. Nesse caso, qualquer pessoa que possa assumir a função poderá criar um cluster de cache. Sua conta da AWS, à qual a função pertence, é a 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 de políticas de permissões.

Note

Esta seção discute o uso do IAM no contexto do Amazon ElastiCache. Não são fornecidas informações detalhadas sobre o serviço IAM. Para obter a documentação completa do IAM, consulte [O que é o IAM?](#) no Guia do usuário do IAM. Para obter mais informações sobre a sintaxe e as descrições da política do IAM, consulte a [Referência de políticas do AWS IAM](#) no Guia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas como políticas baseadas em identidade (políticas do IAM). As políticas anexadas a um recurso são chamadas de políticas baseadas em recursos.

Tópicos

- [Políticas baseadas em identidade \(políticas do IAM\)](#)
- [Especificar elementos da política: ações, efeitos, recursos e entidades principais](#)
- [Especificar condições em uma política](#)

Políticas baseadas em identidade (políticas do IAM)

Você pode anexar políticas a identidades do IAM. Por exemplo, você pode fazer o seguinte:

- Anexar uma política de permissões a um usuário ou grupo na sua conta: um administrador de conta pode usar uma política de permissões associada a determinado usuário para conceder permissões. Nesse caso, as permissões são para o usuário criar um recurso do ElastiCache, como um cluster de cache, um grupo de parâmetros ou um grupo de segurança.
- Anexar uma política de permissões a uma função: você pode anexar uma política de permissões baseada em identidade a um perfil do IAM para conceder permissões entre contas. Por exemplo, o administrador na conta A pode criar uma função para conceder permissões entre contas para outra conta da AWS (por exemplo, conta B) ou um serviço da AWS, conforme o seguinte:
 1. Um administrador da Conta A cria uma função do IAM e anexa uma política de permissões à função que concede permissões em recursos da Conta A.
 2. Um administrador da Conta A anexa uma política de confiança à função identificando a Conta B como a entidade principal, que pode assumir a função.
 3. Depois, o administrador da Conta B pode delegar permissões para assumir a função a todos os usuários na Conta B. Isso permite que os usuários na Conta B criem ou acessem recursos na

Conta A. Em alguns casos, convém conceder permissões a um serviço da AWS para assumir a função. Para oferecer suporte a essa abordagem, o principal da política de confiança também pode ser um principal de serviço da AWS.

Para obter mais informações sobre o uso do IAM para delegar permissões, consulte [Gerenciamento de acesso](#) no Guia do usuário do IAM.

A seguir está um exemplo de política que permite ao usuário executar a ação `DescribeCacheClusters` para sua conta da AWS. O ElastiCache também permite identificar recursos específicos usando os ARNs de recurso para as ações da API. 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 mais informações sobre como usar políticas baseadas em identidade com o ElastiCache, consulte [Uso de políticas baseadas em identidade \(políticas do IAM\) para o Amazon ElastiCache](#). Para obter mais informações sobre usuários, grupos, funções e permissões, consulte [Identidades \(usuários, grupos e funções\)](#) no Guia do usuário do IAM.

Especificar elementos da política: ações, efeitos, recursos e entidades principais

Para cada recurso do Amazon ElastiCache (consulte [Recursos e operações do Amazon ElastiCache](#)), o serviço define um conjunto de operações de API (consulte [Ações](#)). Para conceder permissões a essas operações da API, o ElastiCache define um conjunto de ações que podem ser especificadas em uma política. Por exemplo, para o recurso de cluster do ElastiCache, as seguintes ações são definidas: `CreateCacheCluster`, `DeleteCacheCluster`, e `DescribeCacheCluster`. A execução de uma operação de API pode exigir permissões para mais de uma ação.

Estes são os elementos de política mais básicos:

- **Recurso:** em uma política, você usa um nome do recurso da Amazon (ARN) para identificar o recurso a que a política se aplica. Para obter mais informações, consulte [Recursos e operações do Amazon ElastiCache](#).
- **Ação:** você usa palavras-chave de ação para identificar operações de recursos que deseja permitir ou negar. Por exemplo, dependendo do `Effect` especificado, a permissão `elasticache:CreateCacheCluster` consente ou nega que o usuário execute a operação `CreateCacheCluster` do Amazon ElastiCache.
- **Efeito:** você especifica o efeito quando o usuário solicita a ação específica, que pode ser permitir ou negar. Se você não conceder (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar acesso explicitamente a um recurso. Por exemplo, você poderia fazer isso para garantir que um usuário não possa acessar o recurso, mesmo se uma política diferente conceder o acesso.
- **Entidade principal** - em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é a entidade principal implícita. Para as políticas baseadas em recursos, você especifica quais usuários, contas, serviços ou outras entidades deseja que recebam permissões (isso se aplica somente a políticas baseadas em recursos).

Para saber mais sobre a sintaxe e as descrições de políticas do IAM, consulte a [Referência de políticas do AWS IAM da](#) no Guia do usuário do IAM.

Para obter uma tabela que mostra todas as ações de API do Amazon ElastiCache, consulte [Permissões da API do ElastiCache: referência de ações, recursos e condições](#).

Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da política do IAM para especificar as condições de quando uma política deverá entrar em vigor. Por exemplo, é recomendável aplicar uma política somente após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte [Condition](#) no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Para usar chaves de condição específicas do ElastiCache, consulte [Uso de chaves de condição](#). Há chaves de condição em toda a AWS que você pode usar conforme apropriado. Para obter uma lista completa de chaves de toda a AWS, consulte [Available Keys for Conditions](#) no Guia do usuário do IAM.

Políticas gerenciadas pela AWS para o Amazon ElastiCache

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para todos os clientes da AWS usarem. 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 em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em uma política gerenciada pela AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política está vinculada. É mais provável que a AWS atualize uma política gerenciada pela AWS quando um novo AWS service (Serviço da AWS) é lançado ou novas operações de API são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte [AWSPolíticas gerenciadas pela](#) no Manual do usuário do IAM.

Política gerenciada pela AWS: ElastiCacheServiceRolePolicy

Não é possível anexar ElastiCacheServiceRolePolicy às entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o ElastiCache realize ações em seu nome.

Esta política permite que o ElastiCache gerencie recursos da AWS em seu nome, conforme necessário para gerenciar seu cache:

- `ec2`: gerencie recursos de rede do EC2 para serem conectados aos nós de cache, inclusive endpoints da VPC (para caches sem servidor), interfaces de rede elásticas (ENIs) (para clusters autoprotetidos) e grupos de segurança.
- `cloudwatch`: emita dados métricos do serviço para o CloudWatch.
- `outposts`: permita a criação de nós de cache em AWS Outposts.

É possível encontrar a política [ElastiCacheServiceRolePolicy](#) no console do IAM e [ElastiCacheServiceRolePolicy](#) no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonElastiCacheFullAccess

É possível anexar a política AmazonElastiCacheFullAccess a suas identidades do IAM.

Essa política permite que as entidades principais tenham acesso total ao ElastiCache usando o AWS Management Console:

- `elasticache`: acesse todas as APIs.
- `iam`: crie uma função vinculada ao serviço necessária para a operação de serviço.
- `ec2`: descreva os recursos dependentes do EC2 necessários para a criação de cache (VPC, sub-rede, grupo de segurança) e permita a criação de endpoints de VPC (para caches sem servidor).
- `kms`: permita o uso de CMKs gerenciadas pelo cliente na criptografia em repouso.
- `cloudwatch`: permita o acesso às métricas para exibir métricas do ElastiCache 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 o AWS Outposts para criação de cache no console.
- `sns`: usado para preencher tópicos do SNS para funcionalidade de notificação no console.

Você pode encontrar a política [AmazonElastiCacheFullAccess](#) no console do IAM e a política [AmazonElastiCacheFullAccess](#) no Guia de referência de políticas gerenciadas pela AWS.

Política gerenciada pela AWS: AmazonElasticacheReadOnlyAccess

É possível anexar a política AmazonElastiCacheReadOnlyAccess a suas identidades do IAM.

Essa política permite que as entidades principais tenham acesso somente leitura ao ElastiCache usando o AWS Management Console:

- `elasticache`: acesse APIs `Describe` somente leitura.

Você pode encontrar a política [AmazonElastiCacheReadOnlyAccess](#) no console do IAM e a política [AmazonElastiCacheReadOnlyAccess](#) no Guia de referência de políticas gerenciadas pela AWS.

Atualizações do ElastiCache para as políticas gerenciadas pela AWS

Visualize detalhes sobre atualizações de políticas gerenciadas AWS para ElastiCache desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed de RSS na página Histórico do documento do ElastiCache.

Alteração	Descrição	Data
AmazonElastiCacheFullAccess : atualização feita em uma política existente	O ElastiCache adicionou novas permissões para permitir o gerenciamento de caches sem servidor e para permitir o uso de todos os recursos do serviço por meio do console.	27 de novembro de 2023
ElastiCacheServiceRolePolicy : atualização feita em uma política existente	O ElastiCache adicionou novas permissões para permitir o gerenciamento de endpoints da VPC para recursos de cache sem servidor.	27 de novembro de 2023
O ElastiCache começou a rastrear alterações	O ElastiCache começou a monitorar as alterações em suas políticas gerenciadas pela AWS.	9 de fevereiro de 2020

Uso de políticas baseadas em identidade (políticas do IAM) para o Amazon ElastiCache

Este tópico fornece exemplos de políticas baseadas em identidade em que um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções).

Important

Recomendamos que você primeiro leia os tópicos que explicam os conceitos básicos e as opções para gerenciar o acesso aos recursos do Amazon ElastiCache. Para obter mais informações, consulte [Visão geral do gerenciamento de permissões de acesso aos recursos do ElastiCache](#).

As seções neste tópico abrangem o seguinte:

- [Políticas gerenciadas pela AWS para o Amazon ElastiCache](#)
- [Exemplos de política gerenciada pelo cliente](#)

A seguir, um exemplo de uma política de permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowClusterPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache",
        "elasticache:CreateCacheCluster",
        "elasticache:DescribeServerlessCaches",
        "elasticache:DescribeReplicationGroups",
        "elasticache:DescribeCacheClusters",
        "elasticache:ModifyServerlessCache",
        "elasticache:ModifyReplicationGroup",
        "elasticache:ModifyCacheCluster"
      ],
      "Resource": "*"
    }
  ],
}
```

```
{
  "Sid": "AllowUserToPassRole",
  "Effect": "Allow",
  "Action": [ "iam:PassRole" ],
  "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
}
```

A política tem duas instruções:

- A primeira instrução concede permissões para as ações do Amazon ElastiCache (`elasticache:Create*`, `elasticache:Describe*` e `elasticache:Modify*`)
- A segunda instrução concede permissões para a ação do IAM (`iam:PassRole`) no nome da função do IAM especificado no final do valor `Resource`.

A política não especifica o elemento `Principal` porque, em uma política baseada em identidade, a entidade principal que obtém as permissões não é especificada. Quando você anexar uma política a um usuário, o usuário será a entidade principal implícita. Quando você anexa uma política de permissões a um perfil do IAM, a entidade principal identificada na política de confiança do perfil obtém as permissões.

Para obter uma tabela que mostre todas as ações da API do Amazon ElastiCache e os recursos aos quais se aplicam, consulte [Permissões da API do ElastiCache: referência de ações, recursos e condições](#).

Exemplos de política gerenciada pelo cliente

Se você não estiver usando uma política padrão e optar por usar uma política gerenciada personalizada, realize uma destas ações: Você deve ter permissões para chamar `iam:createServiceLinkedRole` (para obter mais informações, consulte [Exemplo 4: permitir a um usuário chamar a API IAM CreateServiceLinkedRole](#)). Ou você deve ter criado uma função vinculada ao serviço do ElastiCache.

Quando combinado com as permissões mínimas necessárias para usar o console do Amazon ElastiCache, as políticas de exemplo nesta seção concedem permissões adicionais. Os exemplos também são relevantes para os SDKs da AWS e para a AWS CLI.

Para obter instruções sobre como configurar usuários e grupos do IAM, consulte [Criação do seu primeiro usuário do IAM e grupo de administradores](#) no Guia do usuário do IAM.

⚠ Important

Sempre teste suas políticas do IAM completamente antes de usá-las em produção. Algumas ações do ElastiCache que parecem simples podem exigir outras ações para oferecer suporte quando você estiver usando o console do ElastiCache. Por exemplo, `elasticache:CreateCacheCluster` concede permissões para criar clusters de cache do ElastiCache. No entanto, para realizar essa operação, o console do ElastiCache usa várias ações `Describe` e `List` para preencher listas de consoles.

Exemplos

- [Exemplo 2: permitir a um usuário acesso somente de leitura a recursos do ElastiCache](#)
- [Exemplo 3: permitir que um usuário realize tarefas comuns de administrador do sistema do ElastiCache](#)
- [Exemplo 4: permitir que um usuário acesse todas as ações da API do ElastiCache](#)
- [Exemplo 4: permitir a um usuário chamar a API IAM `CreateServiceLinkedRole`](#)
- [Exemplo 5: permitir que um usuário se conecte ao cache sem servidor usando a autenticação do IAM](#)

Exemplo 2: permitir a um usuário acesso somente de leitura a recursos do ElastiCache

A seguinte política concede permissões a ações do ElastiCache que permitem que um usuário liste 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 3: permitir que um usuário realize tarefas comuns de administrador do sistema do ElastiCache

Entre as tarefas do administrador do sistema comuns estão a modificação de recursos. Um administrador do sistema também pode querer obter informações sobre eventos do ElastiCache. A seguinte política concede permissões de usuário para executar ações do ElastiCache para essas tarefas comuns de administrador de sistema. Normalmente, você anexa esse tipo de política de permissões ao grupo de administradores do sistema.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "EAllowMutations",
    "Effect": "Allow",
    "Action": [
      "elasticache:Modify*",
      "elasticache:Describe*",
      "elasticache:ResetCacheParameterGroup"
    ],
    "Resource": "*"
  }
]
```

Exemplo 4: permitir que um usuário acesse todas as ações da API do ElastiCache

A seguinte política permite que um usuário acesse todas as ações do ElastiCache. 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 a um usuário chamar a API IAM CreateServiceLinkedRole

A política a seguir permite que o usuário chame a API CreateServiceLinkedRole do IAM. Recomendamos conceder esse tipo de política de permissões para o usuário que invoca operações mutatórias do ElastiCache.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "elasticache.amazonaws.com"
        }
      }
    }
  ]
}
```

Exemplo 5: permitir que um usuário se conecte ao cache sem servidor usando a autenticação do IAM

A política a seguir permite que qualquer usuário se conecte a qualquer cache sem servidor usando a autenticação do IAM entre 2023-04-01 e 2023-06-30.

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
      ],
      "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},

```

```
    "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
  },
  {
    "Effect" : "Allow",
    "Action" : ["elasticache:Connect"],
    "Resource" : [
      "arn:aws:elasticache:us-east-1:123456789012:user:*"
    ]
  }
]
```

Permissões em nível de recurso

Você pode restringir o escopo das permissões especificando recursos em uma política do IAM. Muitas ações de API do ElastiCache oferecem suporte a um tipo de recurso que varia de acordo com o comportamento da ação. Cada instrução de política do IAM concede permissão a uma ação realizada em um recurso. Quando a ação não atua em um recurso indicado, ou quando você concede permissão para executar a ação em todos os recursos, o valor do recurso na política é um curinga (*). Para muitas ações de API, restrinja os recursos que um usuário pode modificar especificando o Amazon Resource Name (ARN – Nome de recurso da Amazon) de um recurso ou um padrão de ARN correspondente a vários recursos. Para restringir as permissões por recurso especifique o recurso por ARN.

Para ver uma lista dos tipos de recursos do ElastiCache e seus ARNs, consulte [Resources Defined by Amazon ElastiCache](#) na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Actions Defined by Amazon ElastiCache](#).

Exemplos

- [Exemplo 1: permitir que um usuário tenha acesso total a tipos de recurso específicos do ElastiCache](#)
- [Exemplo 2: negue a um usuário acesso a um cache sem servidor.](#)

Exemplo 1: permitir que um usuário tenha acesso total a tipos de recurso específicos do ElastiCache

A seguinte política permite explicitamente todos os recursos do tipo cache sem servidor.

```
{
```

```
"Sid": "Example1",
"Effect": "Allow",
"Action": "elasticache:*",
"Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
]
}
```

Exemplo 2: negue a um usuário acesso a um cache sem servidor.

O exemplo a seguir nega explicitamente o acesso a um determinado cache sem servidor.

```
{
  "Sid": "Example2",
  "Effect": "Deny",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
  ]
}
```

Uso de chaves de condição

Você pode especificar as condições que determinam como uma política do IAM entra em vigor. No ElastiCache, é possível usar o elemento `Condition` de uma política JSON para comparar chaves no contexto da solicitação com os valores de chave especificados em sua política. Para obter mais informações, consulte [Elementos da política JSON do IAM: condição](#).

Para ver uma lista de chaves de condição do ElastiCache, consulte [Condition Keys for 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, grave uma política de permissões do IAM que especifique condições para controlar um conjunto de parâmetros individuais em determinadas solicitações. Em seguida, aplique a política aos usuários, grupos ou funções do IAM que você criar usando o console do IAM.

Para aplicar uma condição, adicione as informações da condição à declaração de política do IAM. No exemplo a seguir, você especifica a condição de que qualquer cluster de cache autoprotetido criado será do tipo de nó cache .r5.large.

```
{
  "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"
          ]
        }
      }
    }
  ]
}
```

Para obter mais informações, consulte [Exemplos de políticas de controle de acesso baseada em tags](#).

Para obter mais informações sobre a utilização de operadores de condição de política, consulte [Permissões da API do ElastiCache: referência de ações, recursos e condições](#).

Políticas de exemplo: uso de condições para controle de acesso refinado

Esta seção mostra políticas de exemplo para a implementação de um controle de acesso refinado nos parâmetros do ElastiCache listados anteriormente.

1. `elasticache:maximumDataStorage`: especifique o armazenamento de dados máximo 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 de ECPU por segundo de um cache sem servidor. Usando as condições fornecidas, o cliente não pode criar caches que possam executar mais do que um número específico de ECPUs por segundo.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:MaximumECPUperSecond": "100000"
        }
      }
    }
  ]
}

```

3. `elasticache:CacheNodeType`: especifique quais `NodeType`(s) 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:NumNodeGroups`: 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"
        }
      }
    }
  ]
}

```

5. elasticache: ReplicaSperNodeGroup: 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"
        }
      }
    }
  ]
}

```

6. elasticache:EngineVersion: especifique o uso do mecanismo versão 5.0.6.

```

{
  "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"
        }
    }
}
]
}

```

7. elasticache:EngineType: especifique usando somente o mecanismo do Redis.

```

{
  "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"
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

8. `elasticache:AtRestEncryptionEnabled`: especifique que grupos de replicação seriam criados somente com criptografia habilitada.

```

{
  "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"
        }
      }
    }
  ]
}

```

9. `elasticache:TransitEncryptionEnabled`

- a. Defina a chave de condição `elasticache:TransitEncryptionEnabled` como `false` para a ação [CreateReplicationGroup](#) a fim de especificar que grupos de replicação só podem ser criados quando o TLS não está sendo usado:

```
{
  "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 condição `elasticache:TransitEncryptionEnabled` é definida como `false` em uma política para a ação [CreateReplicationGroup](#), uma solicitação `CreateReplicationGroup` será permitida somente se o TLS não estiver sendo usado (ou seja, se a solicitação não incluir um parâmetro `TransitEncryptionEnabled` definido como `true` ou um parâmetro `TransitEncryptionMode` definido como `required`).

- b. Defina a chave de condição `elasticache:TransitEncryptionEnabled` como `true` para a ação [CreateReplicationGroup](#) a fim de especificar que grupos de replicação só podem ser criados quando o TLS está sendo usado:

```
{
  "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 condição `elasticache:TransitEncryptionEnabled` é definida como `true` em uma política para a ação [CreateReplicationGroup](#), uma solicitação `CreateReplicationGroup` será permitida somente se incluir um parâmetro `TransitEncryptionEnabled` definido como `true` e um parâmetro `TransitEncryptionMode` definido como `required`.

- c. Defina `elasticache:TransitEncryptionEnabled` como `true` para a ação `ModifyReplicationGroup` a fim de especificar que os grupos de replicação só podem ser modificados quando o TLS está sendo usado:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:ModifyReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "BoolIfExists": {
          "elasticache:TransitEncryptionEnabled": "true"
        }
      }
    }
  ]
}
```

Quando a chave de condição `elasticache:TransitEncryptionEnabled` é definida como `true` em uma política para a ação [ModifyReplicationGroup](#), uma solicitação `ModifyReplicationGroup` será permitida somente se incluir um parâmetro `TransitEncryptionMode` definido como `required`. Opcionalmente, o parâmetro `TransitEncryptionEnabled` definido como `true` também pode ser incluído, mas não é necessário nesse caso para habilitar o TLS.

10. `elasticache:AutomaticFailOverEnabled`: especifique que grupos de replicação sejam criados somente com failover automático habilitado.

```
{
  "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"
      }
    }
  }
]
}

```

11 `elasticache:MultiAZEnabled`: especifique que os grupos de replicação não possam ser criados com o multi-AZ desabilitado.

```

{
  "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"
      }
    }
  }
]
}

```

12 `elasticache:ClusterModeEnabled`: especifique que grupos de replicação só possam ser criados com o modo cluster habilitado.

```

{
  "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"
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

13. `elasticache:AuthTokenEnabled`: especifique que grupos de replicação só possam ser criados com o token AUTH habilitado.

```

{
  "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"
        }
      }
    }
  ]
}

```

14. `elasticache:SnapshotsRetentionLimit`: especifique o número de dias (ou mínimo/máximo) para manter o snapshot. 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"
        }
      }
    }
  ]
}
```

15. `elasticache:KmsKeyId`: especifique o uso de chaves do KMS da AWS gerenciadas pelo 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"
        }
      }
    }
  ]
}

```

16 `elasticache:CacheParameterGroupName`: 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. Em seguida há um exemplo restringindo 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"
        }
      }
    }
  ]
}

```

17 `elasticache:CreateCacheCluster`: 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"
            ]
        }
    }
}
]
}

```

18. `elasticache:CacheNodeType`: permissão de `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"
        }
      }
    }
  ]
}

```

Note

Ao criar políticas para impor tags e outras chaves de condição juntas, o `IfExists` condicional pode ser necessário na condição de elementos-chave devido aos requisitos

extras de política `elasticache:AddTagsToResource` para solicitações de criação com o parâmetro `--tags`.

Uso de funções vinculadas ao serviço para o Amazon ElastiCache

O Amazon ElastiCache usa [funções vinculadas ao serviço](#) do AWS Identity and Access Management IAM. A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a um serviço da AWS, como o Amazon ElastiCache. As funções vinculadas ao serviço do Amazon ElastiCache são predefinidas pelo Amazon ElastiCache. Elas incluem todas as permissões que o serviço exige para chamar os serviços da AWS em nome dos seus clusters.

Uma função vinculada ao serviço facilita a configuração do Amazon ElastiCache porque você não precisa adicionar as permissões necessárias manualmente. As funções ainda existem dentro de sua contada AWS mas estão vinculadas aos casos de uso do Amazon ElastiCache e têm permissões predefinidas. Somente o Amazon ElastiCache pode assumir essas funções e somente essas funções podem usar a política de permissões predefinidas. Você pode excluir os perfis somente depois de primeiro excluir seus recursos relacionados. Isso protege seus recursos do Amazon ElastiCache, pois você não pode remover por engano as permissões necessárias para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que apresentam Yes (Sim) na coluna Service-Linked Role (Função vinculada a serviço). Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Sumário

- [Permissões de função vinculada ao serviço para o Amazon ElastiCache](#)
 - [Permissões para criar uma função vinculada ao serviço](#)
- [Criação de uma função vinculada ao serviço \(IAM\)](#)
 - [Criação de uma função vinculada ao serviço \(console do IAM\)](#)
 - [Criação de uma função vinculada ao serviço \(CLI do IAM\)](#)
 - [Criação de uma função vinculada ao serviço \(API do IAM\)](#)
- [Edição da descrição de uma função vinculada a serviço para o Amazon ElastiCache](#)
 - [Edição da descrição de uma função vinculada ao serviço \(console do IAM\)](#)
 - [Edição da descrição de uma função vinculada ao serviço \(CLI do IAM\)](#)
 - [Edição da descrição de uma função vinculada ao serviço \(API do IAM\)](#)

- [Exclusão de uma função vinculada ao serviço do Amazon ElastiCache](#)
 - [Limpar uma função vinculada ao serviço](#)
 - [Exclusão de uma função vinculada ao serviço \(console do IAM\)](#)
 - [Exclusão de uma função vinculada ao serviço \(CLI do IAM\)](#)
 - [Exclusã de uma função vinculada ao serviço \(API do IAM\)](#)

Permissões de função vinculada ao serviço para o Amazon ElastiCache

Permissões para criar uma função vinculada ao serviço

Para permitir que uma entidade do IAM crie uma função vinculada ao serviço ServiceRoleForElastiCache da AWS

Adicione a seguinte declaração de política às permissões dessa entidade IAM:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole",
    "iam:PutRolePolicy"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWSServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

Para permitir que uma entidade do IAM exclua uma função vinculada ao serviço ServiceRoleForElastiCache da AWS

Adicione a seguinte declaração de política às permissões dessa entidade IAM:

```
{
  "Effect": "Allow",
  "Action": [
    "iam>DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWSServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

```
}
```

Como alternativa, você pode usar uma política gerenciada pela AWS para fornecer acesso total ao Amazon ElastiCache.

Criação de uma função vinculada ao serviço (IAM)

Você pode criar uma função vinculada ao serviço usando o console do IAM, a CLI ou a API.

Criação de uma função vinculada ao serviço (console do IAM)

Você pode usar o console do IAM para criar uma função vinculada ao serviço.

Para criar uma função vinculada ao serviço (console)

1. Faça login em AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Perfis. Em seguida, escolha Criar nova função.
3. Em Select type of trusted entity (Selecionar tipo de entidade confiável), selecione Serviço da AWS.
4. Em Or select a service to view its use cases (Ou selecione um serviço para visualizar seus casos de uso), escolha ElastiCache.
5. Escolha Next: Permissions (Próximo: permissões).
6. Em Nome da política, observe que `ElastiCacheServiceRolePolicy` é necessário para esta função. Escolha Próximo: tags.
7. Observe que não há suporte para as tags para funções vinculadas ao serviço. Escolha Próximo: análise.
8. (Opcional) Em Descrição da função, edite a descrição para a nova função vinculada ao serviço.
9. Revise a função e escolha Criar função.

Criação de uma função vinculada ao serviço (CLI do IAM)

Você pode usar operações do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço. Essa função pode incluir a política de confiança e as políticas em linha de que o serviço precisa para assumir a função.

Para criar uma função vinculada ao serviço (CLI)

Use a seguinte operação:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

Criação de uma função vinculada ao serviço (API do IAM)

Você pode usar a API do IAM para excluir uma função vinculada ao serviço. Essa função pode conter a política de confiança e as políticas em linha de que o serviço precisa para assumir a função.

Para criar uma função vinculada ao serviço (API)

Use a chamada da API [CreateServiceLinkedRole](#). Na solicitação, especifique o nome do serviço na forma de `elasticache.amazonaws.com`.

Edição da descrição de uma função vinculada a serviço para o Amazon ElastiCache

O Amazon ElastiCache não permite que você edite a função vinculada ao serviço `ServiceRoleForElastiCache` da AWS. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ela. No entanto, será possível editar a descrição do perfil usando o IAM.

Edição da descrição de uma função vinculada ao serviço (console do IAM)

Também é possível usar o console do IAM para editar a descrição de uma função vinculada ao serviço.

Para editar a descrição de uma função vinculada ao serviço (console)

1. No painel de navegação do console do IAM, escolha Perfis.
2. Escolha o nome da função a ser modificada.
3. No extremo direito da Descrição da função, escolha Editar.
4. Insira uma nova descrição na caixa e escolha Salvar.

Edição da descrição de uma função vinculada ao serviço (CLI do IAM)

Você pode usar operações do IAM na AWS Command Line Interface para editar a descrição de uma função vinculada ao serviço.

Para alterar a descrição de uma função (CLI)

1. (Opcional) Para visualizar a descrição atual de uma função, use a AWS CLI para a operação do IAM [get-role](#).

Example

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

Use o nome da função, não o nome de recurso da Amazon (ARN), para fazer referência às funções com as operações da CLI. Por exemplo, se uma função tiver o seguinte nome de recurso da Amazon (ARN): `arn:aws:iam::123456789012:role/myrole`, você fará referência à função como **myrole**.

2. Para atualizar a descrição de uma função vinculada ao serviço, use a AWS CLI para a operação do IAM [update-role-description](#).

Para Linux, macOS ou Unix:

```
$ aws iam update-role-description \  
  --role-name AWSServiceRoleForElastiCache \  
  --description "new description"
```

Para Windows:

```
$ aws iam update-role-description ^  
  --role-name AWSServiceRoleForElastiCache ^  
  --description "new description"
```

Edição da descrição de uma função vinculada ao serviço (API do IAM)

Você pode usar a API do IAM para editar uma descrição de função vinculada ao serviço.

Para alterar a descrição de uma função (API)

1. (Opcional) Para visualizar a descrição atual de uma função, use a operação da API do IAM [GetRole](#).

Example

```
https://iam.amazonaws.com/  
?Action=GetRole  
&RoleName=AWSServiceRoleForElastiCache  
&Version=2010-05-08  
&AUTHPARAMS
```

2. Para atualizar uma descrição de função, use a operação da API do IAM [UpdateRoleDescription](#).

Example

```
https://iam.amazonaws.com/  
?Action=UpdateRoleDescription  
&RoleName=AWSServiceRoleForElastiCache  
&Version=2010-05-08  
&Description="New description"
```

Exclusão de uma função vinculada ao serviço do Amazon ElastiCache

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar sua função vinculada ao serviço antes de excluí-la.

O Amazon ElastiCache não exclui a função vinculada ao serviço para você.

Limpar uma função vinculada ao serviço

Antes de usar o IAM para excluir uma função vinculada ao serviço, primeiro confirme que a função não tem recursos (clusters ou grupos de replicação) associados a ela.

Para verificar se a função vinculada ao serviço tem uma sessão ativa no console do IAM

1. Faça login em AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Perfis. Então, escolha o nome (não a caixa de seleção) da função ServiceRoleForElastiCache da AWS.
3. Na página Resumo para a função selecionada, escolha a guia Consultor de Acesso.

4. Na guia Consultor de Acesso, revise a atividade recente para a função vinculada ao serviço.

Para excluir recursos do Amazon ElastiCache que exigem ServiceRoleForelastiCache da AWS

- Para excluir um cluster, consulte o seguinte:
 - [Usar a AWS Management Console](#)
 - [Usar a AWS CLI](#)
 - [Uso da API do ElastiCache](#)
- Para excluir um grupo de replicação, consulte o seguinte:
 - [Exclusão de um grupo de replicação \(Console\)](#)
 - [Excluir um grupo de replicação \(AWS CLI\)](#)
 - [Exclusão de um grupo de replicação \(API do ElastiCache\)](#)

Exclusão de uma função vinculada ao serviço (console do IAM)

É possível usar o console do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (console)

1. Faça login em AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Roles. Selecione a caixa de marcação ao lado do nome da função que você deseja excluir, não o nome ou a linha em si.
3. Em ações de Role (Função) na parte superior da página, escolha a função Delete (Excluir).
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço que mostram quando cada uma das funções selecionadas acessou pela última vez um produto da AWS. Isso ajuda você a confirmar se a função está ativo no momento. Se quiser prosseguir, escolha Sim, Excluir para enviar a função vinculada ao serviço para exclusão.
5. Monitore as notificações do console do IAM para progresso da exclusão da função vinculada ao serviço. Como a exclusão da função vinculada ao serviço do IAM é assíncrona, depois de enviar a função para exclusão, a tarefa pode ou não ser bem-sucedida. Se a tarefa obtiver êxito, você poderá escolher Visualizar Detalhes ou Visualizar Recursos a partir das notificações para saber por que a exclusão falhou.

Exclusão de uma função vinculada ao serviço (CLI do IAM)

Você pode usar operações do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço.

Para excluir uma função vinculado ao serviço (CLI)

1. Se você não souber o nome da função vinculada ao serviço que deseja excluir, insira o seguinte comando. Esse comando lista as funções e os nomes de recursos da Amazon (ARNs) em sua conta.

```
$ aws iam get-role --role-name role-name
```

Use o nome da função, não o nome de recurso da Amazon (ARN), para fazer referência às funções com as operações da CLI. Por exemplo, se uma função tiver o ARN `arn:aws:iam::123456789012:role/myrole`, você fará referência à função como **myrole**.

2. Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `deletion-task-id` da resposta para verificar o status da tarefa de exclusão. Insira o seguinte para enviar uma solicitação de exclusão de função vinculada ao serviço.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Insira o seguinte para verificar o estado da tarefa de exclusão.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Exclusã de uma função vinculada ao serviço (API do IAM)

É possível usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (API)

1. Para enviar uma solicitação de exclusão de um roll vinculada ao serviço, chame [DeleteServiceLinkedRole](#). Na solicitação, especifique o nome da função.

Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `DeletionTaskId` da resposta para verificar o status da tarefa de exclusão.

2. Para verificar o status da exclusão, chame [GetServiceLinkedRoleDeletionStatus](#). Na solicitação, especifique o `DeletionTaskId`.

O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Permissões da API do ElastiCache: referência de ações, recursos e condições

Ao configurar o [controle de acesso](#) e escrever políticas de permissões para anexar a uma política do IAM (baseada em identidade ou baseada em recurso), use a tabela a seguir como referência. A tabela lista cada operação da API do Amazon ElastiCache e as ações correspondentes para as quais você pode conceder permissões para executar a ação. Você especifica as ações no campo `Action` da política e um valor de recurso no campo `Resource` da política. Salvo indicação em contrário, o recurso é obrigatório. Alguns campos incluem um recurso obrigatório e recursos opcionais. Quando não há ARN de recurso, o recurso na política é um caractere curinga (*).

Você pode usar as chaves de condição em toda a nas suas políticas do ElastiCache para expressar condições. Para ver uma lista de chaves de condição específicas do ElastiCache, juntamente com as ações e os tipos de recursos aos quais elas se aplicam, consulte [Uso de chaves de condição](#). Para obter uma lista completa de chaves em toda a AWS, consulte [Chaves de contexto de condição globais da AWS](#) no Guia do usuário do IAM.

Note

Para especificar uma ação, use o prefixo `elasticache:` seguido do nome da operação da API (por exemplo, `elasticache:DescribeCacheClusters`).

Para ver uma lista das ações do ElastiCache, consulte [Actions Defined by Amazon ElastiCache](#) na Referência de autorização do serviço.

Validação de conformidade para o Amazon ElastiCache

Audidores externos avaliam a segurança e a conformidade dos serviços da AWS como parte de vários programas de conformidade da AWS, como SOC, PCI, FedRAMP e HIPAA.

Para saber se um AWS service (Serviço da AWS) está no escopo de programas de conformidade específicos, consulte [Serviços da AWS no escopo por programa de conformidade](#) em que você está interessado. Para obter informações gerais, consulte [AWS Programas de conformidade](#).

É possível fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte [Downloading Reports in AWS Artifact](#).

Sua responsabilidade de conformidade ao usar o Serviços da AWS é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos

regulamentos e leis aplicáveis. A AWS fornece os seguintes atributos para ajudar com a conformidade:

- [Guias de referência rápida de conformidade e segurança](#) - estes guias de implantação discutem considerações sobre arquitetura e fornecem as etapas para a implantação de ambientes de linha de base focados em segurança e conformidade na AWS.
- [Arquitetura para segurança e conformidade com HIPAA no Amazon Web Services](#): esse whitepaper descreve como as empresas podem usar a AWS para criar aplicações adequadas aos padrões HIPAA.

 Note

Nem todos os Serviços da AWS estão qualificados pela HIPAA. Para obter mais informações, consulte a [Referência dos serviços qualificados pela HIPAA](#).

- [atributos de conformidade da AWS](#): essa coleção de manuais e guias pode ser aplicada a seu setor e local.
- [Guias de conformidade do cliente da AWS](#): entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as práticas recomendadas para proteção de 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)).
- [Avaliar atributos com regras](#) no AWS Config Guia do desenvolvedor: o serviço AWS Config avalia como as configurações de atributos estão em conformidade com práticas internas, diretrizes do setor e regulamentos.
- [AWS Security Hub](#): este AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança na AWS. O Security Hub usa controles de segurança para avaliar os atributos 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](#).
- [AWS Audit Manager](#): esse AWS service (Serviço da AWS) ajuda a auditar continuamente seu uso da para simplificar a forma como você gerencia os riscos e a conformidade com regulamentos e padrões do setor.

Mais informações

Para obter informações gerais sobre a conformidade com a Nuvem AWS, consulte o seguinte:

- [Endpoints FIPS por serviço](#)
- [Atualizações de serviço em ElastiCache](#)
- [AWS Compatibilidade da nuvem](#)
- [Modelo de responsabilidade compartilhada](#)
- [Programa de conformidade da AWS com o PCI DSS](#)

Resiliência no Amazon ElastiCache

A infraestrutura global da AWS é criada com base em regiões da AWS e zonas de disponibilidade. As regiões da AWS fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, as quais são conectadas com baixa latência, alto throughput e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte [Infraestrutura global da AWS](#).

Além da infraestrutura global da AWS, o Amazon ElastiCache oferece vários recursos para ajudar a dar suporte às necessidades de resiliência de dados e backup.

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 Redis](#)

- [Recomendações](#)

Mitigar falhas ao executar o Redis

Ao executar o mecanismo Redis, você tem as opções a seguir para minimizar o impacto de um nó ou uma falha 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 autoprojetoados devem ser devidamente configurados para mitigar a falha de um nó individual.

Para mitigar o impacto de falhas em nós Redis em clusters autoprojetoados, você tem as seguintes opções:

Tópicos

- [Mitigar falhas: grupos de replicação Redis](#)

Mitigar falhas: grupos de replicação Redis

Um grupo de replicação Redis é composto por um único nó primário no/do qual seu aplicativo pode ler e gravar e de 1 a 5 nós de réplica somente leitura. Sempre que os dados são gravados no nó primário, eles também são atualizados de maneira assíncrona nos nós de réplica de leitura.

Quando uma réplica de leitura falha

1. ElastiCache detecta a falha na réplica de leitura.
2. ElastiCache desativa o nó com falha.
3. ElastiCache lança e provisiona um nó de substituição na mesma AZ.
4. O novo nó é sincronizado com o nó primário.

Durante esse período, seu aplicativo pode continuar lendo e gravando usando os outros nós.

Multi-AZ do Redis

É possível habilitar o Multi-AZ em seus grupos de replicação do Redis. 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 ter mais informações, consulte [Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ](#).

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 proteção adicional, recomendamos que você inicie os nós no 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 ter 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 de qualquer AZ resultará apenas na perda dos nós nessa AZ.

Sempre que você perde um nó, pode passar por uma degradação do desempenho, pois as operações de leitura são compartilhadas por menos nós. Essa degradação do desempenho continuará até que os nós sejam substituídos.

Para obter informações sobre como especificar as zonas de disponibilidade para nós Redis, consulte [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#).

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Escolher regiões e zonas de disponibilidade](#).

Recomendações

É recomendável criar caches sem servidor em clusters autoprotetidos, pois você obtém automaticamente uma melhor tolerância a falhas sem configuração adicional. No entanto, durante a criação de um cluster autoprotetido, 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, nós recomendamos que sua implementação use vários nós em cada estilhaço e distribua esses nós em várias zonas de disponibilidade. Isso é feito automaticamente para caches sem servidor.

Para clusters autoprotetidos, 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.

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. Espalhar nós de forma

uniforme em AZs minimizará o impacto no evento improvável de uma falha de AZ. Isso é feito automaticamente para caches sem servidor.

Outras precauções

Se você estiver executando o Redis, além da precaução acima, recomendamos que você programe backups regulares do seu cluster. Backups (snapshots) criam um arquivo .rdb que você pode usar para restaurar seu cache em caso de falha ou corrupção. Para ter mais informações, consulte [Snapshots e restauração](#).

Segurança da infraestrutura no ElastiCache da AWS

Como um serviço gerenciado, o ElastiCache da AWS é protegido pelos procedimentos de segurança de rede global da AWS descritos na seção Segurança e conformidade do [Centro de Arquitetura da AWS](#).

Você usa chamadas de API publicadas pela AWS para acessar o ElastiCache por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Atualizações de serviço em ElastiCache

ElastiCache monitora automaticamente sua frota de caches, clusters e nós para aplicar atualizações de serviço à medida que elas se tornam disponíveis. As atualizações de serviço para caches sem servidor são aplicadas de maneira transparente e automática. Para clusters autoprojettados, 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 autoprotetidos. Você também pode monitorar o progresso dessas atualizações no ElastiCache cluster selecionado em tempo real.

Gerenciando atualizações de serviços

ElastiCache atualizações de serviço para clusters autoprotetidos são lançadas regularmente. Se você tiver um ou mais clusters autoprotetidos qualificados para essas atualizações de serviço, receberá notificações por e-mail, SNS, Personal Health Dashboard (PHD) e 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.

Tópicos

- [Como aplicar as atualizações de serviço](#)
- [Verificando se você tem a atualização de serviço mais recente aplicada usando o console AWS](#)
- [Interromper as atualizações do serviço](#)

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, a API ou AWS CLI aplicar a atualização. As seções a seguir explicam as opções que você pode usar para aplicar as atualizações.

Aplicação de atualizações de serviço usando o console

Para visualizar a lista de atualizações de serviço disponíveis, além de outras informações, acesse a página Atualizações de serviço no console.

1. Faça login 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 ElastiCache clusters do Redis estão executando a atualização de serviço mais recente seguindo estas etapas:

1. Escolha um cluster aplicável na página Redis 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-b...	Status	Cluster ...
elasticache-redis-6-2-6-update-202310109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 03:59:59 (UT...	Available	March 1, 2...

Se o console exibir “Nenhuma atualização de serviço encontrada”, significa que o cluster ElastiCache for Redis já tem a atualização de serviço mais recente aplicada.

Service update name	Cluster...	Update type	Update severity	Release date	Recommended ...
No service updates found.					

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 ter 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 no Redis ElastiCache

A lista de Vulnerabilidades e exposições comuns (CVE) é uma lista de vulnerabilidades de segurança cibernética publicamente conhecidas. Cada entrada é um link que contém um número

de identificação, uma descrição e pelo menos uma referência pública. Você pode encontrar nesta página uma lista de vulnerabilidades de segurança que foram abordadas no ElastiCache Redis.

Recomendamos que você sempre atualize para a versão mais recente ElastiCache do Redis para se proteger contra vulnerabilidades conhecidas. Ao operar um cache ElastiCache sem servidor, as correções de CVE são aplicadas automaticamente ao seu cache. Ao operar clusters autoprojeto, o ElastiCache for Redis expõe o componente PATCH. Por exemplo, ao usar ElastiCache a versão 6.2.6 do Redis, a versão principal é 6, a versão secundária é 2 e a versão do patch é 6. As versões PATCH são para correções de bugs compatíveis com versões anteriores, correções de segurança e alterações não funcionais.

Você pode usar esta página para verificar se uma versão específica do ElastiCache for Redis tem uma correção para uma vulnerabilidade de segurança específica. Se o seu cluster ElastiCache for Redis estiver executando uma versão sem a correção de segurança, consulte a tabela abaixo e tome medidas. Você pode atualizar para uma versão mais recente ElastiCache para Redis contendo a correção ou, se estiver em uma versão ElastiCache para Redis contendo a correção, verifique se a atualização de serviço mais recente foi aplicada consultando a [Gerenciando atualizações de serviços](#). Para obter mais informações sobre as versões compatíveis com ElastiCache o mecanismo Redis e como fazer o upgrade, consulte [Versões de mecanismos e atualização](#).

Note

- Se um CVE for endereçado em uma versão ElastiCache para Redis, isso significa que ele também é abordado nas versões mais recentes. Então, por exemplo, se uma vulnerabilidade for resolvida na ElastiCache versão 6.0.5 do Redis, 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 for Redis executando a versão ElastiCache for Redis 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 para Redis na qual seu cluster está sendo executado, consulte [Gerenciando atualizações de serviços](#).

ElastiCache para a versão Redis	CVEs abordados
Redis 6.0.5	CVE-2022-24735 * , CVE-2022-24736 *

ElastiCache para a versão Redis	CVEs abordados
Redis 6.2.6	CVE-2022-24834 * , CVE-2022-35977 * , CVE-2022-36021* , CVE-2022-24735 , CVE-2022-24736
Redis 7.0.7	CVE-2023-41056* , CVE-2022-24834* , CVE-2022-35977 , CVE-2022-36021 , CVE-2022-24735 , CVE-2022-24736
Redis 7.1.0	CVE-2023-41056 , CVE-2022-24834 , CVE-2022-35977 , CVE-2022-36021 , CVE-2022-24735 , CVE-2022-24736

Registros e monitoramento no Amazon ElastiCache

Para gerenciar o cache, é importante que você saiba o desempenho dos caches. O ElastiCache gera métricas publicadas no Amazon CloudWatch Logs para monitorar o desempenho do cache. Além disso, o ElastiCache gera eventos quando mudanças significativas acontecem nos recursos de cache (por exemplo, um novo cache é criado ou um cache é excluído).

Tópicos

- [Métricas e eventos sem servidor](#)
- [Métricas e eventos de clusters autoprotetidos](#)
- [Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail](#)

Métricas e eventos sem servidor

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

O namespace AWS/ElastiCache inclui as métricas do CloudWatch a seguir para os caches sem servidor do Redis.

Métrica	Descrição	Unidade
BytesUsedForCache	O número total de bytes usados pelos dados armazenados no cache.	Bytes
ElastiCacheProcessingUnits	O número total de ElastiCacheProcessingUnits (eCPUs)	Contagem

Métrica	Descrição	Unidade
	consumidas pelas solicitações executadas no cache	
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
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 cuja utilização foi controlada pelo ElastiCache porque a workload estava escalando mais rápido do que o ElastiCache pode escalar.	Contagem

Métrica	Descrição	Unidade
NewConnections	O número total de conexões que foram aceitas pelo servidor durante esse período.	Contagem
Currltems	O número de itens no cache.	Contagem
CurrVolatileItems	O número de itens no cache com TTL.	Contagem
NetworkBytesIn	Total de bytes transferidos para dentro do cache	Bytes
NetworkBytesOut	Total de bytes transferidos para fora do cache	Bytes
Evictions	A contagem de chaves removidas pelo cache	Contagem
IamAuthenticationExpirations	O número total de conexões do Redis expiradas autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
IamAuthenticationThrottling	O número total de solicitações restritas do Redis AUTH ou HELLO autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem

Métrica	Descrição	Unidade
KeyAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar chaves que eles não têm permissão para acessar. Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
AuthenticationFailures	O número total de tentativas falhadas de autenticação para Redis 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

O ElastiCache também emite as métricas no nível de comando a seguir. Para cada tipo de comando, o ElastiCache emite a contagem total de comandos e o número de eCPUs consumidas por esse tipo de comando.

Métrica	Descrição	Unidade
EvalBasedCmds	O número de comandos get recebidos pelo cache.	Contagem

Métrica	Descrição	Unidade
EvalBasedCmdsECPUs	ECPUs consumidas por comandos baseados em avaliação.	Contagem
GeoSpatialBasedCmds	O número total de comandos para comandos baseados em dados geoespaciais. Isso é derivado da estatística <code>commandstats</code> do Redis. Ele é derivado somando todos o tipos de comandos <code>geo</code> : <code>geoadd</code> , <code>geodist</code> , <code>geohash</code> , <code>geopos</code> , <code>georadius</code> e <code>georadiusbymember</code> .	Contagem
GeoSpatialBasedCmdsECPUs	ECPUs consumidas por comandos geoespaciais.	Contagem
GetTypeCmds	O número total de comandos tipo somente leitura. Isso é derivado da estatística <code>commandstats</code> do Redis somando todos os comandos do tipo somente leitura (<code>get</code> , <code>hget</code> , <code>scard</code> , <code>lrange</code> etc.)	Contagem
GetTypeCmdsECPUs	ECPUs consumidas por comandos de leitura.	Contagem

Métrica	Descrição	Unidade
HashBasedCmds	O número total de comandos baseados em hash. Isso é derivado da estatística <code>commandstats</code> do Redis somando todos os comandos que atuam em um ou mais hashes (<code>hget</code> , <code>hkeys</code> , <code>hvals</code> , <code>hdel</code> etc.).	Contagem
HashBasedCmdsECPUs	ECPUs consumidas por comandos baseados em hash.	Contagem
HyperLogLogBasedCmds	O número total de comandos baseados em HyperLogLog. Isso é derivado da estatística <code>commandstats</code> do Redis somando todos os comandos do tipo <code>pf</code> (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> etc.).	Contagem
HyperLogLogBasedCmdsECPUs	ECPUs consumidas por comandos baseados em HyperLogLog.	Contagem
JsonBasedCmds	O número total de comandos JSON, incluindo comandos de leitura e gravação. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em chaves JSON.	Contagem
JsonBasedCmdsECPUs	ECPUs consumidas por todos os comandos JSON, incluindo comandos de leitura e gravação.	Contagem

Métrica	Descrição	Unidade
JsonBasedGetCmds	O número total de comandos JSON somente leitura. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos de leitura que atuam em chaves JSON.	Contagem
JsonBasedGetCmdsECPUs	eCPUs consumidas por comandos JSON somente leitura.	Contagem
JsonBasedSetCmds	O número total de comandos de gravação JSON. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos de gravação que atuam em chaves JSON.	Contagem
JsonBasedSetCmdsECPUs	eCPUs consumidas por comandos JSON de leitura.	Contagem
KeyBasedCmds	O número total de comandos baseados em chave. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em uma ou mais chaves em várias estruturas de dados (<code>del</code> , <code>expire</code> , <code>rename</code> etc.).	Contagem
KeyBasedCmdsECPUs	ECPUs consumidas por comandos baseados em chave.	Contagem

Métrica	Descrição	Unidade
ListBasedCmds	O número total de comandos baseados em lista. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em uma ou mais listas (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> etc.).	Contagem
ListBasedCmdsECPUs	ECPUs consumidas por comandos baseados em lista.	Contagem
NonKeyTypeCmds	O número total de comandos não baseados em chave. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos que não atuam em uma chave; por exemplo, <code>acl</code> , <code>dbsize</code> ou <code>info</code> .	Contagem
NonKeyTypeCmdsECPUs	ECPUs consumidas por comandos não baseados em chave.	Contagem
PubSubBasedCmds	O número total de comandos para a funcionalidade pub/sub. Isso é derivado de <code>commandstats</code> do Redis somando todos os comandos usados para a funcionalidade pub/sub: <code>psubscribe</code> , <code>ssubscribe</code> , <code>spublish</code> , <code>subscribe</code> , <code>ssubscribe</code> e <code>unsubscribe</code> .	Contagem

Métrica	Descrição	Unidade
PubSubBasedCmdsECPUs	ECPUs consumidas por comandos baseados em pub/sub.	Contagem
SetBasedCmds	O número total de comandos que são baseados em conjuntos. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em um ou mais conjuntos (<code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>sunion</code> etc.).	Contagem
SetBasedCmdsECPUs	ECPUs consumidas por comandos baseados em conjunto.	Contagem
SetTypeCmds	O número total de tipos de comando de gravação. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os tipos de comando mutative que operam em dados (<code>set</code> , <code>hset</code> , <code>sadd</code> , <code>lpop</code> etc.)	Contagem
SetTypeCmdsECPUs	eCPUs consumidas por comandos de leitura.	Contagem

Métrica	Descrição	Unidade
SortedSetBasedCmds	O número total de comandos que são classificados com base em conjuntos. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em um ou mais conjuntos classificados (<code>zcount</code> , <code>zrange</code> , <code>zrank</code> , <code>zadd</code> etc.).	Contagem
SortedSetBasedCmdsECPUs	ECPUs consumidas por comandos baseados em classificação.	Contagem
StringBasedCmds	O número total de comandos baseados em string. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em uma ou mais strings (<code>strlen</code> , <code>setex</code> , <code>setrange</code> etc.).	Contagem
StringBasedCmdsECPUs	ECPUs consumidas por comandos baseados em string.	Contagem

Métrica	Descrição	Unidade
StreamBasedCmds	O número total de comandos que são baseados em fluxo. Isso é derivado da estatística commandstats do Redis, somando todos os comandos que atuam em um ou mais tipos de dados de fluxos (xrange, xlen, xadd, xdel etc.).	Contagem
StreamBasedCmdsECPUs	ECPUs consumidas por comandos baseados em fluxo.	Contagem

Eventos de cache sem servidor

O ElastiCache registra eventos de logs relacionados ao 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 log usando o console do ElastiCache, o comando describe-events da AWS CLI ou a ação de API DescribeEvents do ElastiCache.

É possível optar por monitorar, ingerir, transformar e agir em eventos do ElastiCache usando o Amazon EventBridge. Saiba mais no Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

Visualização de eventos do ElastiCache (console)

Para visualizar eventos usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha Eventos.
3. Na tela Eventos, cada linha da lista representa um evento e exibe a origem do evento, o tipo de evento, o horário GMT do evento e uma descrição do evento. Usando a opção Filtro, você pode especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualização de eventos do ElastiCache (AWS CLI)

Para gerar uma lista de eventos do ElastiCache usando a 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 cache está pronto para ser usado.

Detail-Type	Descrição	Unidade	Origem	Message
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Endereços IP livres insuficientes para criar um endpoint da VPC.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Sub-redes inválidas fornecidas na solicitação.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Limite de cota atingido para criação de um endpoint da VPC.
Falha na criação de cache	Fio de cache	falha	serverless-cache	Falha na criação do cache <cache-name>. Você não tem permissões para criar um endpoint da VPC.

Detail-Type	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 versão do Redis incompatível 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 <user-group-name> fornecido 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 <cache-name>. Houve falha na restauração de dados a partir do snapshot porque <reason>.</p> <p>Motivos de falha:</p> <ul style="list-style-type: none"> • falhou na recuperação do arquivo do S3. • O md5 esperado não corresponde ao md5 real. • O arquivo RDB fornecido tem uma versão não compatível.

Eventos de atualização de cache sem servidor

Detail-Type	Lista de recursos	Categoria	Origem	Message
Cache atualizado	Fio de cache	alteração de configuração	serverless-cache	SecurityGroups atualizados 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 versão do Redis incompatível está presente no grupo de usuários <user-group-name>.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	Falha na atualização do cache <cache-name>. Falha na atualização do SecurityGroups.
Falha na atualização do cache	Fio de cache	alteração de configuração	serverless-cache	Falha na atualização do cache <cache-name>. Falha na atualização do SecurityGroups por causa de 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>. Falha na atualização dos SecurityGroups porque os SecurityGroups são inválidos.

Eventos de exclusão do cache sem servidor

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 do cache sem servidor

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 snapshot de cache sem servidor

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 <cache-name>.</p> <p>Falha na criação do snapshot <snapshot-name> com a chave gerenciada pelo cliente <key-id> <reason>.</p> <p>Mensagens de motivo da falha:</p> <ul style="list-style-type: none"> • porque a chave gerenciada pelo cliente está desabilitada • porque não foi possível encontrar a chave gerenciada pelo cliente

Detail-Type	Resources-list	Categoria	Origem	Message
				<ul style="list-style-type: none"> porque a solicitação atingiu o tempo limite
Snapshot creation failed	Fio de cache Fio de snapshot	falha	serverless-cache-snapshot	<p>Falha na criação do snapshot para o cache <cache-name>. Houve falha na criação do snapshot <snapshot-name> por causa de <reason>.</p> <p>Motivo padrão:</p> <ul style="list-style-type: none"> por causa de um erro interno
Falha na exportação do snapshot	Fio de snapshot	falha	serverless-cache-snapshot	<p>Falha na exportação do snapshot para o cache <cache-name>. Não foi possível exportar o snapshot para o bucket %s porque o 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 clusters autoprotetidos

Esta seção descreve as métricas, os eventos e os logs que você pode ver ao trabalhar com clusters autoprotetidos.

Tópicos

- [Métricas para clusters autoprotetidos](#)

- [Eventos para clusters autoprojeto](#)
- [Entrega de logs](#)
- [Monitoramento do uso com métricas do CloudWatch](#)
- [Monitoramento de eventos do ElastiCache pelo Amazon SNS](#)

Métricas para clusters autoprojeto

Quando você projeta clusters autoprojeto, o ElastiCache emite métricas em cada nível de nó, inclusive métricas no 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 Redis](#).

Eventos para clusters autoprojeto

O ElastiCache registra eventos de logs relacionados aos caches autoprojeto. Ao trabalhar com clusters autoprojeto, você pode exibir os eventos de cluster no console do ElastiCache, usando a AWS CLI ou usando o Amazon Simple Notification Service (SNS). Eventos de cluster autoprojeto não são publicados no 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 log usando o console do ElastiCache, o comando `describe-events` da AWS CLI ou a ação de API `DescribeEvents` do ElastiCache.

Visualização de eventos do ElastiCache (console)

O procedimento a seguir exibe eventos usando o console do ElastiCache.

Para visualizar eventos usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha **Events** (Eventos).
3. Na tela **Eventos**, cada linha da lista representa um evento e exibe a origem do evento, o tipo de evento, o horário GMT do evento e uma descrição do evento. Usando a opção **Filter**, você pode

especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualização de eventos do ElastiCache (AWS CLI)

Para gerar uma lista de eventos do ElastiCache usando a 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 seguintes eventos do ElastiCache acionam notificações do Amazon SNS. Para obter informações sobre detalhes de eventos, consulte [Visualização de eventos do ElastiCache](#).

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 a endereços IP livres insuficientes	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.

Nome do evento	Message	Descrição
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 a estado de rede incompatível	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Foi feita uma tentativa de executar um novo cluster de cache em uma nuvem privada virtual (VPC) inexistente.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Escalabilidade para cluster de cache concluída com êxito.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	A operação de expansão no cluster de cache falhou.

Nome do evento	Message	Descrição
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>Um dos seguintes eventos ocorreu:</p> <ul style="list-style-type: none">• A lista de security groups de cache autorizados para o cluster de cache foi modificada.• Um ou mais novos security groups do EC2 foram autorizados em qualquer um dos security groups de cache associados ao cluster de cache.• Um ou mais security groups do EC2 foram revogados de qualquer um dos security groups de cache associados ao cluster de cache.

Nome do evento	Message	Descrição
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>O ElastiCache detectou que o host que está executando um nó de cache está degradado ou é inacessível e iniciou a substituição do nó de cache.</p> <div data-bbox="1068 493 1507 808"><p> Note</p><p>A entrada de DNS para o nó de cache substituído não é 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 de clientes de cache podem parar de usar o nó de cache mesmo após o ElastiCache ter substituído o nó de cache. Nesse caso, a aplicação deve atualizar a lista de servidores quando esse evento ocorrer.</p>

Nome do evento	Message	Descrição
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>O ElastiCache detectou que o host que executa um nó de cache está degradado ou é inacessível e concluiu a substituição do nó de cache.</p> <div data-bbox="1068 495 1507 806"><p> Note</p><p>A entrada de DNS para o nó de cache substituído não é 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 de clientes de cache podem parar de usar o nó de cache mesmo após o ElastiCache ter substituído o nó de cache. Nesse caso, a aplicação deve atualizar a lista de servidores quando esse evento ocorrer.</p>

Nome do evento	Message	Descrição
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Um ou mais nós de cache foram reinicializados. Mensagem (Memcached): "Cache node %s shutdown" Em seguida, uma segunda mensagem: "Cache node %s restarted"
ElastiCache:CertificateRenewalComplete (somente Redis)	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 Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	O failover para um nó de réplica foi bem-sucedido.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	O número de réplicas no cluster foi aumentado.

Nome do evento	Message	Descrição
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	O processo de adição de réplicas ao cluster começou.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Um nó no seu cluster que estava programado para substituição já não está programado para substituição.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Um nó no seu cluster previamente programado para substituição foi reprogramado para substituição durante a nova janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós .
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Um nó no seu cluster está programado para substituição durante a janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Um nó de cache foi removido do cluster de cache.

Nome do evento	Message	Descrição
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	A operação de expansão no grupo de replicação foi concluída com sucesso.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	A operação de expansão no grupo de replicação falhou.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Uma atualização de autoatendimento está disponível para o nó.
ElastiCache:SnapshotComplete (somente Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Um snapshot de cache foi concluído com sucesso.
ElastiCache:SnapshotFailed (somente Redis)	SnapshotFailed : <i>cluster-name</i>	Um snapshot de cache falhou. Consulte os eventos de cache do cluster para mais uma causa detalhada. Se você descrever o snapshot, consulte DescribeSnapshots , o status será failed.

Entrega de logs

Note

O Redis Slow Log é compatível com os clusters de cache e grupos de replicação do Redis usando o mecanismo da versão 6.0 em diante.

O Redis Engine Log é compatível com os clusters de cache e grupos de replicação do Redis usando o mecanismo da versão 6.2 em diante.

A entrega de logs permite que você transmita Redis [SLOWLOG](#) ou Redis 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 log será entregue ao destino especificado em um dos dois formatos: JSON ou TEXT.

Um número fixo de entradas de log lento são recuperados periodicamente do mecanismo do Redis. 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 APIs de modificação, ou [modify-cache-cluster](#). [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 ElastiCache for Redis 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 for Redis 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 log do ElastiCache for Redis Engine 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("#")`.
- `Time (Hora)` - O horário UTC da mensagem registrada. A hora está no seguinte formato: `"DD MMM YYYY hh:mm:ss.ms UTC"`
- `Role (Função)` - Função do nó de onde o log é emitido. Pode ser um dos seguintes: `"M"` para primário, `"S"` para réplica, `"C"` para um processo secundário de gravação que está trabalhando em RDB/AOF ou `"X"` para sentinela.
- `Message (Mensagem)` - Mensagem de log do Redis Engine.

Permissões para configurar os logs

Você precisa incluir as seguintes permissões do IAM em sua política de usuário/função do IAM:

- `logs:CreateLogDelivery`

- logs:UpdateLogDelivery
- logs>DeleteLogDelivery
- logs:GetLogDelivery
- logs>ListLogDeliveries

Para obter mais informações, consulte [Visão geral do gerenciamento de acesso: permissões e políticas](#).

Especificações de tipo de log e formato de log

Log lento

O log lento suporta JSON e TEXT

O exemplo a seguir mostra um exemplo de formato JSON.

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

O exemplo a seguir mostra um exemplo de formato TEXT.

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Log do mecanismo

O log do mecanismo é compatível com JSON e TEXT

O exemplo a seguir mostra um exemplo de formato JSON.

```
{
  "CacheClusterId": "xxxxxxxxzy-engine-log-test",
```

```
"CacheNodeId": "0001",  
"LogLevel": "VERBOSE",  
"Role": "M",  
"Time": "12 Nov 2020 01:28:57.994 UTC",  
"Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.  
Check back later"  
}
```

O exemplo a seguir mostra um exemplo de formato TEXT.

```
xxxxxxxxxxxxzy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua  
script detected that is still in execution after 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 logs de vários clusters do Redis e grupos de replicação podem ser entregues no mesmo grupo de logs.
- 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 deve ter as seguintes configurações de permissões para configurar o Redis ElastiCache para enviar 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 logs de vários clusters do Redis e grupos de replicação podem ser entregues ao mesmo fluxo 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 Redis ElastiCache para enviar registros 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

Ao usar o AWS Management Console, você pode criar um cluster do Redis (modo cluster desabilitado) seguindo as etapas em [Criação de um cluster do Redis \(modo cluster desabilitado\) \(console\)](#) ou criar um cluster do Redis (modo cluster habilitado) usando as etapas em [Criação de um cluster do Redis \(modo cluster habilitado\) \(console\)](#). Em ambos os casos, você configura a entrega de logs fazendo o seguinte:

1. Em Advanced Redis settings (Configurações avançadas do Redis), escolha Logs e, em seguida, Slow logs (Logs lentos) ou Engine logs (Logs de mecanismos).
2. Em Log format (Formato do log), escolha Text (Texto) ou JSON.
3. Em Destination Type (Tipo de destino), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.
4. Em Log destination (Destino do log), escolha Create new (Criar) e insira o nome do bucket do Amazon S3, o nome do grupo de logs do CloudWatchLogs ou o nome do fluxo do Kinesis Data Firehose, ou escolha Select existing (Selecionar existente) e, em seguida, escolha o nome do grupo de logs do CloudWatchLogs ou o nome do fluxo do Kinesis Data Firehose,

Ao modificar um cluster:

Você pode optar por habilitar/desabilitar a entrega de logs ou alterar o tipo de destino, o formato ou o destino:

1. Faça login no console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No painel de navegação, escolha Redis clusters (Clusters do Redis).
3. Na lista de clusters, escolha o cluster que você deseja modificar. Selecione o Cluster name (Nome do cluster), e não a caixa de seleção ao lado dele.
4. Na página Cluster name (Nome do cluster), escolha a guia Logs.

5. Para habilitar/desabilitar logs lentos, escolha Enable slow logs (Habilitar logs lentos) ou Disable slow logs (Desabilitar logs lentos).
6. Para ativar/desativar os logs do mecanismo, escolha Enable engine logs (Ativar logs do mecanismo) ou Disable engine logs (Desativar logs do mecanismo).
7. Para alterar sua configuração, escolha Modify slow logs (Modificar logs lentos) ou Modify engine logs (Modificar logs do mecanismo):
 - Em Destination Type (Tipo de destino), escolha CloudWatch Logs (Logs do CloudWatch) ou Kinesis Firehose.
 - Em Log destination (Destino do log), escolha Create new (Criar novo) e insira o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose. Ou escolha Select existing (Selecionar existente) e escolha o nome do grupo de logs do CloudWatchLogs ou o nome do stream do Kinesis Data Firehose.

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 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
}'
```

Monitoramento do uso com métricas do CloudWatch

O ElastiCache fornece métricas que permitem monitorar seus clusters. Você pode acessar essas métricas através do CloudWatch. Para obter mais informações sobre o CloudWatch, consulte a [Documentação do CloudWatch](#).

O ElastiCache fornece métricas em nível de host (por exemplo, uso da CPU) e métricas que são específicas do software do mecanismo de cache (por exemplo, obtenções e ausências no cache). Essas métricas são medidas e publicadas para cada nó de cache em intervalos de 60 segundos.

Important

Você deve considerar a configuração de alarmes do CloudWatch em determinadas métricas-chave, para ser notificado caso a performance do cluster de cache começar a se degradar. Para obter mais informações, consulte [Quais métricas devo monitorar?](#) neste guia.

Tópicos

- [Métricas em nível de host](#)
- [Métricas para Redis](#)
- [Quais métricas devo monitorar?](#)
- [Escolher estatísticas e períodos de métricas](#)
- [Monitoramento de cluster do CloudWatch e métricas de nó](#)

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 Redis](#)

Métrica	Descrição	Unidade
<code>CPUUtilization</code>	A porcentagem de utilização da CPU para o host inteiro. Como o Redis é de thread único, recomendamos que você monitore a métrica <code>EngineCPUUtilization</code> , para nós com quatro ou mais vCPUs.	Percentual
<code>CPUCreditBalance</code>	<p>O número de créditos ganhos de CPU que uma instância acumulou desde que foi executada ou iniciada. Para a T2 Padrão, o <code>CPUCreditBalance</code> também inclui o número de créditos de execução 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, 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 do <code>CPUCreditBalance</code> são disponibilizados para que a instância gaste e apresente intermitência com uma utilização de CPU acima da linha de base.</p> <p>As métricas de crédito de CPU estão disponíveis a uma frequência de apenas 5 minutos.</p> <p>Essa métrica não está disponível para instâncias expansíveis T2.</p>	Créditos (minutos de vCPU)

Métrica	Descrição	Unidade
CPUCreditUsage	<p>O número de créditos de CPU gastos pela instância por utilização de CPU. Um crédito de CPU equivale a um vCPU em execução em 100% de utilização por um minuto ou a uma combinação equivalente de vCPUs, utilização e tempo (por exemplo, um vCPU em execução a 50% de utilização por dois minutos ou dois vCPUs em execução a 25% de utilização por dois minutos).</p> <p>As métricas de crédito de CPU estão disponíveis a uma frequência de apenas 5 minutos. Se você especificar um período de mais cinco minutos, use a estatística Sum (Soma) em vez da estatística Average (Média).</p> <p>Essa métrica não está disponível para instâncias expansíveis T2.</p>	Créditos (minutos de vCPU)
FreeableMemory	A quantidade de memória livre disponível no host. Ela é derivada da RAM, dos buffers e do cache que o sistema operacional relata como passíveis de liberação.	Bytes
NetworkBytesIn	O número de bytes que o host leu da rede.	Bytes
NetworkBytesOut	O número de bytes enviados em todas as interfaces de rede pela instância.	Bytes
NetworkPacketsIn	O número de pacotes recebidos em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de entrada em termos do número de pacotes em uma única instância.	Contagem

Métrica	Descrição	Unidade
NetworkPacketsOut	O número de pacotes enviados em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de saída em termos do número de pacotes em uma única instância.	Contagem
NetworkBandwidthInAllowanceExceeded	Número de pacotes na fila ou descartados porque a largura de banda agregada de entrada excedeu o máximo para a instância.	Contagem
NetworkConntrackAllowanceExceeded	Número de pacotes descartados porque o monitoramento da conexão excedeu o máximo para a instância e não foi possível estabelecer novas conexões. Isso pode resultar em perda de pacotes para tráfego indo para a instância ou vindo da instância	Contagem
NetworkBandwidthOutAllowanceExceeded	Número de pacotes na fila ou descartados porque a largura de banda agregada de saída excedeu o máximo para a instância.	Contagem
NetworkPacketsPerSecondAllowanceExceeded	O número de pacotes na fila ou descartados porque o valor bidirecional de pacotes por segundo excedeu o máximo para a instância.	Contagem
NetworkMaxBytesIn	A intermitência máxima de bytes recebidos em cada minuto.	Bytes
NetworkMaxBytesOut	A intermitência máxima de bytes transmitidos em cada minuto.	Bytes
NetworkMaxPacketsIn	A intermitência máxima de pacotes recebidos em cada minuto.	Contagem
NetworkMaxPacketsOut	A intermitência máxima de pacotes transmitidos em cada minuto.	Contagem

Métrica	Descrição	Unidade
SwapUsage	A quantidade de troca usada no host.	Bytes

Métricas para Redis

O namespace AWS/ElastiCache inclui as métricas de Redis a seguir.

Com exceção de `ReplicationLag` e `EngineCPUUtilization`, essas métricas são derivadas do comando `info` do Redis. Cada métrica é calculada no nível de nó de cache.

Para a documentação completa do comando `info` do Redis, consulte <http://redis.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 desfragmação ativo. Deriva da estatística <code>active_defrag_hits</code> no comando INFO do Redis .	Número
AuthenticationFailures	O número total de tentativas falhadas de autenticação para Redis usando o comando <code>AUTH</code> . É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
BytesUsedForCache	O número total de bytes alocados pelo Redis para todas as finalidades, inclusive o conjunto de dados, os buffers, etc.	Bytes
	Dimension: <code>Tier=Memory</code> para clusters do Redis usando Classificação de dados	Bytes

Métrica	Descrição	Unidade
	<p>em níveis: o número total de bytes usados para cache por memória. Esse é o valor de <code>used_memory</code> estática em INFO do Redis.</p> <p>Dimension: Tier=SSD para clusters do Redis usando Classificação de dados em níveis: o número total de bytes usados para cache por SSD.</p>	Bytes
BytesReadFromDisk	O número total de bytes lidos no disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis .	Bytes
BytesWrittenToDisk	O número total de bytes gravados no disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis .	Bytes
CacheHits	O número de buscas de chaves somente leitura bem-sucedidas no dicionário principal. Deriva da estatística <code>keyspace_hits</code> no comando INFO do Redis .	Contagem
CacheMisses	O número de buscas de chaves somente leitura malsucedidas no dicionário principal. Deriva da estatística <code>keyspace_misses</code> no comando INFO do Redis .	Contagem
CommandAuthorizationFailures	O número total de tentativas falhadas por usuários para executar comandos que eles não têm permissão para chamar. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem

Métrica	Descrição	Unidade
CacheHitRate	Indica a eficiência de uso da instância Redis. 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
ChannelAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar canais que eles não têm permissão para acessar. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme nesta métrica para detectar tentativas de acesso não autorizadas.	Contagem
CurrConnections	O número de conexões de clientes, excluindo conexões de réplicas de leitura. O ElastiCache usa de duas a quatro conexões para monitorar o cluster em cada caso. Deriva da estatística <code>connected_clients</code> no comando INFO do Redis .	Contagem
CurrItems	O número de itens no cache. Deriva da estatística <code>keyspace</code> do Redis, somando todas as chaves em todo o <code>keyspace</code> .	Contagem
	Dimension: Tier=Memory para clusters usando Classificação de dados em níveis . O número de itens em memória.	Contagem

Métrica	Descrição	Unidade
	Dimension: Tier=SSD (unidades de estado sólido) para clusters usando Classificação de dados em níveis . O número de itens em SSD.	Contagem
CurrVolatileItems	Número total de chaves em todos os bancos de dados que têm um ttl definido. Deriva da estatística <code>expires</code> do Redis somando todas as chaves em todo o keyspace.	Contagem
DatabaseCapacityUsagePercentage	<p>Porcentagem da capacidade total de dados para o cluster que está em uso. Essa métrica é calculada como:</p> $\text{used_memory}/\text{maxmemory}$ <p>Em instâncias de dados em camadas, a métrica é calculada como:</p> $(\text{used_memory} - \text{mem_not_counted_for_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})$ <p>onde <code>used_memory</code> e <code>maxmemory</code> são retirados do Redis INFO.</p>	Percentual

Métrica	Descrição	Unidade
DatabaseCapacityUsageCountedForEvictPercentage	<p>Porcentagem da capacidade total de dados para o cluster que está em uso, excluindo a memória usada para sobrecarga e COB. Essa métrica é calculada como:</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>onde <code>used_memory</code> e <code>maxmemory</code> são retirados do Redis INFO</p>	Percentual
DatabaseMemoryUsagePercentage	<p>Porcentagem de memória para o cluster que está em uso. Isso é calculado usando <code>used_memory/maxmemory</code> de Redis INFO.</p>	Percentual
DatabaseMemoryUsageCountedForEvictPercentage	<p>Porcentagem de memória para o 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> de Redis INFO.</p>	Percentual

Métrica	Descrição	Unidade
DB0AverageTTL	Expõe o <code>avg_ttl</code> de DBO a partir da estatística <code>keyspace</code> do comando INFO do Redis . As réplicas não expiram as chaves; em vez disso, esperam que os nós primários expirem as chaves. Quando um nó primário expira uma chave (ou a remove por causa do LRU), ele sintetiza um DEL, que é transmitido para todas as réplicas. Assim sendo, DB0AverageTTL é 0 para nós de réplica, devido ao fato de que eles não expiram chaves e, portanto, não rastreiam TTL.	Milissegundos

Métrica	Descrição	Unidade
EngineCPUUtilization	Fornece utilização de CPU do thread de mecanismo do Redis. Como o Redis é de thread único, você pode usar essa métrica para analisar a carga do próprio processo do Redis. A métrica EngineCPUUtilization fornece uma visibilidade mais precisa do processo do Redis. Você pode usá-la em conjunto com a métrica CPUUtilization. CPUUtilization expõe a utilização de CPU da instância do servidor como um todo, incluindo outros processos de sistema operacional e de gerenciamento. Para tipos de nós maiores com quatro vCPUs ou mais, use a métrica EngineCPUUtilization para monitorar e definir limites para a escalabilidade.	Percentual

 Note

Em um host do ElastiCache, há processos em segundo plano monitorando o host para fornecer uma experiência de banco de dados gerenciado. Esses processos em segundo plano podem ocupar uma parte significativa da workload da CPU. Isso não é significativo em hosts maiores com mais de duas vCPUs. Mas pode afetar hosts menores com 2vCPUs ou menos. Se você monitorar apenas a métrica EngineCPUUtilization, desconhecerá situações em que o host está sobrecarregado com o alto uso da CPU do Redis e o alto uso da CPU dos processos de monitoramento em segundo plano.

Métrica	Descrição	Unidade
	<p>Portanto, recomendamos monitorar a métrica <code>CPUUtilization</code> para hosts com duas vCPUs ou menos.</p>	
Evictions	O número de chaves que foram removidas devido ao limite <code>maxmemory</code> . Deriva da estatística <code>evicted_keys</code> no comando INFO do Redis .	Contagem
GlobalDatastoreReplicationLag	Esse é o atraso entre o nó primário da região secundária e o nó primário da região primária. Para o Redis habilitado pelo modo de cluster, o atraso indica o atraso máximo entre os fragmentos.	Segundos
IamAuthenticationExpirations	O número total de conexões do Redis expiradas autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
IamAuthenticationThrottling	O número total de solicitações restritas do Redis AUTH ou HELLO autenticadas pelo IAM. Você pode encontrar mais informações sobre o Autenticação com o IAM no guia do usuário.	Contagem
IsMaster	Indica se o nó é o nó primário do fragmento /cluster atual. A métrica pode ser 0 (não primária) ou 1 (primária).	Contagem

Métrica	Descrição	Unidade
KeyAuthorizationFailures	O número total de tentativas falhadas por usuários de acessar chaves que eles não têm permissão para acessar. É possível encontrar mais informações sobre falhas de autenticação individuais usando o comando ACL LOG . Sugerimos definir um alarme para detectar tentativas de acesso não autorizadas.	Contagem
KeysTracked	O número de chaves que estão sendo monitoradas pelo monitoramento de chaves do Redis como um percentual de <code>tracking-table-max-keys</code> . O monitoramento de chaves é usado para ajudar o cache do lado do cliente e notifica os clientes quando as chaves são modificadas.	Contagem
MemoryFragmentationRatio	Indica a eficiência na alocação de memória do mecanismo Redis. Certos limites significarão comportamentos diferentes. O valor recomendado é ter fragmentação acima de 1,0. Isso é calculado a partir de <code>mem_fragmentation_ratio statistic</code> do comando INFO do Redis .	Número

Métrica	Descrição	Unidade
NewConnections	<p>O número total de conexões que foram aceitas pelo servidor durante esse período. Deriva da estatística <code>total_connections_received</code> no comando INFO do Redis.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Se você estiver usando o ElastiCache para Redis versão 5 ou anterior, entre duas e quatro conexões relacionadas por essa métrica serão usadas pelo ElastiCache para monitorar o cluster. No entanto, ao usar o ElastiCache para Redis versão 6 ou posterior, as conexões usadas pelo ElastiCache para monitorar o cluster não serão incluídas nessa métrica.</p> </div>	Contagem
NumItemsReadFromDisk	O número total de itens recuperados do disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis .	Contagem
NumItemsWrittenToDisk	O número total de itens gravados no disco por minuto. Compatível somente para clusters usando Classificação de dados em níveis .	Contagem
MasterLinkHealthStatus	Esse status tem dois valores: 0 ou 1. O valor 0 indica que os dados no nó primário do ElastiCache não estão sincronizados com o Redis no EC2. O valor de 1 indica que os dados não estão sincronizados. Para concluir a migração, use a operação da API CompleteMigration .	Booleano

Métrica	Descrição	Unidade
Reclaimed	O número total de eventos de expiração de chaves. Deriva da estatística <code>expired_keys</code> no comando INFO do Redis .	Contagem
ReplicationBytes	Para nós em uma configuração replicada, <code>ReplicationBytes</code> informa o número de bytes que a primária está enviando para todas as suas réplicas. Essa métrica é representante da carga de gravação no grupo de replicação. Deriva da estatística <code>master_repl_offset</code> no comando INFO do Redis .	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 principal. Para o mecanismo do Redis versão 5.0.6 e posteriores, o atraso pode ser medido em milissegundos.	Segundos
SaveInProgress	Essa métrica binária retorna 1 sempre que um salvamento de plano de fundo (com ou sem bifurcação) está em andamento e 0 caso não esteja. Um processo de salvamento de plano de fundo geralmente é usado durante snapshots e sincronizações. Essas operações podem prejudicar o desempenho. Usando a métrica <code>SaveInProgress</code> , você pode diagnosticar se uma performance prejudicada foi causada por um processo de salvamento de plano de fundo. Deriva da estatística <code>rdb_bgsave_in_progress</code> no comando INFO do Redis .	Booliano

Métrica	Descrição	Unidade
TrafficManagementActive	<p>Indica se o ElastiCache para Redis está gerenciando ativamente o tráfego ao ajustar o tráfego alocado para comandos de entrada, monitoramento ou replicação. O tráfego é gerenciado quando mais comandos são enviados ao nó do o Redis pode processar 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> Note</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

Disponibilidade da EngineCPUUtilization

As regiões da AWS listadas a seguir estão disponíveis em todos os tipos de nó com suporte.

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	Ásia-Pacífico (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 (Beijing)
cn-northwest-2	China (Ningxia)
me-south-1	Oriente Médio (Bahrein)
eu-central-1	Europa (Frankfurt)
eu-west-1	Europa (Irlanda)
eu-west-2	Europa (Londres)
eu-west-3	UE (Paris)
eu-south-1	Europa (Milão)
af-south-1	África (Cidade do Cabo)
eu-north-1	Europa (Estocolmo)

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 baseadas no tipo de comando, incluindo o número de chamadas, o tempo total de CPU consumido por esses comandos e a CPU média consumida por execução de comando. Para cada tipo de comando, é adicionada a seguinte linha: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

As métricas de latência listadas a seguir são calculadas com a estatística `commandstats` de [Redis INFO](#). Elas são calculadas da seguinte maneira: $\text{delta}(\text{usec})/\text{delta}(\text{calls})$. O `delta` é calculado como a diferença dentro de um minuto. A latência é definida como o tempo de CPU gasto pelo ElastiCache para processar o comando. Observe que, para clusters que usam classificação de dados por nível, o tempo necessário para buscar itens no SSD não está incluído nessas medições.

Para obter uma lista completa dos comandos disponíveis, consulte [comandos do redis](#) na documentação do Redis.

Métrica	Descrição	Unidade
<code>ClusterBasedCmds</code>	O número total de comandos que são baseados em cluster. Deriva da estatística <code>commandstats</code> do Redis somando todos os comandos que atuam em um cluster (<code>cluster slot</code> , <code>cluster info</code> etc.).	Contagem
<code>ClusterBasedCmdsLatency</code>	Latência de comandos baseados em cluster.	Microssegundos
<code>EvalBasedCmds</code>	O número total de comandos para comandos baseados em avaliação. Isso é derivado da estatística <code>commandstats</code> do Redis pela soma de <code>eval</code> , <code>evalsha</code> .	Contagem

Métrica	Descrição	Unidade
EvalBasedCmdsLatency	Latência de comandos baseados em avaliação.	Microssegundos
GeoSpatialBasedCmds	O número total de comandos para comandos baseados em dados geoespaciais. Isso é derivado da estatística <code>commandstats</code> do Redis. Ele é derivado somando todos os tipos de comandos geo: <code>geoadd</code> , <code>geodist</code> , <code>geohash</code> , <code>geopos</code> , <code>georadius</code> , e <code>georadiusbymember</code> .	Contagem
GeoSpatialBasedCmdsLatency	Latência de comandos baseados em dados geoespaciais.	Microssegundos
GetTypeCmds	O número total de comandos do tipo read-only. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos do tipo read-only (<code>get</code> , <code>hget</code> , <code>scard</code> , <code>lrange</code> , etc.).	Contagem
GetTypeCmdsLatency	Latência de comandos de leitura.	Microssegundos
HashBasedCmds	O número total de comandos baseados em hash. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em um ou mais hashes (<code>hget</code> , <code>hkeys</code> , <code>hvals</code> , <code>hdel</code> , etc.).	Contagem
HashBasedCmdsLatency	Latência de comandos baseados em hash.	Microssegundos
HyperLogLogBasedCmds	O número total de comandos baseados em HyperLogLog. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos do tipo pf (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> , etc.).	Contagem

Métrica	Descrição	Unidade
HyperLogLogBasedCmdsLatency	Latência de comandos baseados em HyperLogLog.	Microssegundos
JsonBasedCmds	O número total de comandos JSON, incluindo comandos de leitura e gravação. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em chaves JSON.	Contagem
JsonBasedCmdsLatency	Latência de todos os comandos JSON, incluindo comandos de leitura e gravação.	Microssegundos
JsonBasedGetCmds	O número total de comandos JSON somente leitura. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos de leitura que atuam em chaves JSON.	Contagem
JsonBasedGetCmdsLatency	Latência dos comandos JSON somente leitura.	Microssegundos
JsonBasedSetCmds	O número total de comandos de gravação JSON. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos de gravação que atuam em chaves JSON.	Contagem
JsonBasedSetCmdsLatency	Latência dos comandos de gravação JSON.	Microssegundos
KeyBasedCmds	O número total de comandos baseados em chave. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em uma ou mais chaves em várias estruturas de dados (<code>del</code> , <code>expire</code> , <code>rename</code> , etc.).	Contagem

Métrica	Descrição	Unidade
KeyBasedCmdsLatency	Latência de comandos baseados em chave.	Microssegundos
ListBasedCmds	O número total de comandos baseados em lista. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em uma ou mais listas (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> , etc.).	Contagem
ListBasedCmdsLatency	Latência de comandos baseados em lista.	Microssegundos
NonKeyTypeCmds	O número total de comandos não baseados em chave. Deriva da estatística <code>commandstats</code> do Redis, somando todos os comandos que não atuam em uma chave; por exemplo, <code>acl</code> , <code>dbsize</code> ou <code>info</code> .	Contagem
NonKeyTypeCmdsLatency	Latência de comandos não baseados em chave.	Microssegundos
PubSubBasedCmds	O número total de comandos para a funcionalidade pub/sub. Isso é derivado das estatísticas <code>commandstats</code> do Redis somando todos os comandos usados para a funcionalidade pub/sub: <code>punsubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>unsubscribe</code> , <code>spublish</code> , <code>subscribe</code> e <code>unsubscribe</code> .	Contagem
PubSubBasedCmdsLatency	Latência de comandos baseados em pub/sub.	Microssegundos

Métrica	Descrição	Unidade
SetBasedCmds	O número total de comandos que são baseados em conjuntos. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em um ou mais conjuntos (<code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>union</code> , etc.).	Contagem
SetBasedCmdsLatency	Latência de comandos baseados em conjunto.	Microsegundos
SetTypeCmds	O número total de tipos de comando write. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os tipos de comando mutative que operam em dados (<code>set</code> , <code>hset</code> , <code>sadd</code> , <code>lpop</code> , etc.).	Contagem
SetTypeCmdsLatency	Latência de comandos de gravação.	Microsegundos
SortedSetBasedCmds	O número total de comandos que são classificados com base em conjuntos. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em um ou mais conjuntos classificados (<code>zcount</code> , <code>zrange</code> , <code>zrank</code> , <code>zadd</code> , etc.).	Contagem
SortedSetBasedCmdsLatency	Latência de comandos baseados em classificação.	Microsegundos
StringBasedCmds	O número total de comandos baseados em string. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em uma ou mais strings (<code>strlen</code> , <code>setex</code> , <code>setrange</code> , etc.).	Contagem

Métrica	Descrição	Unidade
StringBasedCmdsLatency	Latência de comandos baseados em string.	Microssegundos
StreamBasedCmds	O número total de comandos que são baseados em fluxo. Isso é derivado da estatística <code>commandstats</code> do Redis, somando todos os comandos que atuam em um ou mais tipos de dados de fluxos (<code>xrange</code> , <code>xlen</code> , <code>xadd</code> , <code>xdel</code> , etc.).	Contagem
StreamBasedCmdsLatency	Latência de comandos baseados em fluxo.	Microssegundos

Quais métricas devo monitorar?

As seguintes métricas do CloudWatch oferecem uma boa visão sobre a performance do ElastiCache. Na maioria dos casos, recomendamos que você defina alarmes do CloudWatch para essas métricas para poder tomar medidas corretivas antes que problemas de performance ocorram.

Métricas para monitorar

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage](#)
- [Evictions](#)
- [CurrConnections](#)
- [Memória](#)
- [Rede](#)
- [Latência](#)
- [Replicação](#)
- [Gerenciamento de tráfego](#)

CPUUtilization

Essa é uma métrica em nível de host relatada como uma porcentagem. Para obter mais informações, consulte [Métricas em nível de host](#).

Para tipos de nós menores com 2 vCPUs ou menos, use a métrica `CPUUtilization` para monitorar a workload.

De modo geral, sugerimos que você defina o limite para 90% da CPU disponível. Como o Redis é 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 do Redis (modo cluster desabilitado): aumente a escala vertical usando um tipo de instância com cache maior.
- Clusters do Redis (modo cluster habilitado): adicione mais fragmentos para distribuir a workload de gravação entre nós primários adicionais.

Tip

Em vez de usar a métrica `CPUUtilization` de nível de host, os usuários do Redis podem usar a métrica de Redis `EngineCPUUtilization`, que informa a porcentagem de uso no núcleo do mecanismo do Redis. Para ver se essa métrica está disponível nos seus nós e obter mais informações, consulte [Métricas do Redis](#).

Para tipos de nós maiores com 4 vCPUs ou mais, talvez você queira usar a métrica `EngineCPUUtilization`, que informa a porcentagem de uso no núcleo do mecanismo do Redis. Para ver se essa métrica está disponível nos seus nós e obter mais informações, consulte [Métricas do Redis](#).

EngineCPUUtilization

Para tipos de nós maiores com 4 vCPUs ou mais, talvez você queira usar a métrica `EngineCPUUtilization`, que informa a porcentagem de uso no núcleo do mecanismo do Redis. Para ver se essa métrica está disponível nos seus nós e obter mais informações, consulte [Métricas do Redis](#).

Para obter mais informações, consulte a seção CPUs em [Monitoramento das práticas recomendadas com o Amazon ElastiCache for Redis usando o Amazon CloudWatch](#).

SwapUsage

Esta é uma métrica em nível de host relatada em bytes. Para obter mais informações, consulte [Métricas em nível de host](#).

Quando a métrica `FreeableMemory` do CloudWatch estiver próxima de 0 (isto é, abaixo de 100 MB) ou a métrica `SwapUsage` for superior à métrica `FreeableMemory`, isso indica que a memória de um nó está sob pressão. Se isso ocorrer, consulte os seguintes tópicos:

- [Garantia de que você tenha memória suficiente para criar um snapshot do Redis](#)

- [Gerenciamento da memória reservada](#)

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.

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 em seu aplicativo; você precisará investigar o comportamento do aplicativo para resolver esse problema.

Para obter mais informações, consulte a seção Conexões em [Monitoramento das práticas recomendadas com o Amazon ElastiCache for Redis usando o Amazon CloudWatch](#).

Memória

A memória é um aspecto central do Redis. Compreender a utilização da memória do seu cluster é necessário para evitar a perda de dados e acomodar o crescimento futuro do seu conjunto de dados. Estatísticas sobre a utilização de memória de um nó estão disponíveis na seção de memória do comando [INFO](#) do Redis.

Para obter mais informações, consulte a seção Memória em [Monitoramento das práticas recomendadas com o Amazon ElastiCache for Redis usando o Amazon CloudWatch](#).

Rede

Um dos fatores determinantes para a capacidade de largura de banda de rede do cluster é o tipo de nó selecionado. Para obter mais informações sobre a capacidade de rede do seu nó, consulte [Definição de preços do Amazon ElastiCache](#).

Para obter mais informações, consulte a seção Rede em [Monitoramento das práticas recomendadas com o Amazon ElastiCache for Redis usando o Amazon CloudWatch](#).

Latência

Você pode medir a latência de um comando com um conjunto de métricas do CloudWatch que fornecem latências agregadas por estrutura de dados. As métricas de latência listadas abaixo são calculadas com a estatística `commandstats` do comando [INFO](#) do Redis.

Para obter mais informações, consulte a seção Latência em [Monitoramento das práticas recomendadas com o Amazon ElastiCache for Redis usando o 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 práticas recomendadas com o Amazon ElastiCache for Redis usando o Amazon CloudWatch](#).

Gerenciamento de tráfego

O ElastiCache para Redis gerencia automaticamente o tráfego em um nó quando são enviados mais comandos de entrada ao nó do que o Redis pode processar. Isso é feito para manter a operação e estabilidade ideais do mecanismo.

Quando o tráfego for gerenciado ativamente em um nó, a métrica `TrafficManagementActive` emitirá pontos de dados de 1. Isso indica que o nó está subdimensionado para a workload que está sendo fornecida. Se essa métrica permanecer em 1 por longos períodos, avalie o cluster para decidir se é necessário aumentar a escala verticalmente ou horizontalmente.

Para obter mais informações, consulte a métrica `TrafficManagementActive` na página [Métricas](#).

Escolher estatísticas e períodos de métricas

Embora o CloudWatch permita que você escolha qualquer estatística e período para cada métrica, nem todas as combinações são úteis. Por exemplo, as estatísticas Average, Minimum e Maximum para CPUUtilization são úteis, mas a estatística Sum não é.

Todas as amostras do ElastiCache são publicadas por uma duração de 60 segundos para cada nó de cache individual. Para qualquer período de 60 segundos, uma métrica de nó de cache apenas conterá uma única amostra.

Para obter mais informações sobre como recuperar métricas para seus nós de cache, consulte [Monitoramento de cluster do CloudWatch e métricas de nó](#).

Monitoramento de cluster do CloudWatch e métricas de nó

O ElastiCache e o CloudWatch estão integrados para que você possa reunir uma variedade de métricas. Você pode monitorar essas métricas usando o CloudWatch.

Note

Os seguintes exemplos requerem ferramentas de linha de comando do CloudWatch. Para mais informações sobre o CloudWatch e para baixar as ferramentas do desenvolvedor, consulte a [página de produto do CloudWatch](#).

Os procedimentos a seguir mostram como usar o CloudWatch para coletar estatísticas de espaço de armazenamento para um cluster de cache durante a 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 os limites do ElastiCache, consulte [Limites de serviço da AWS](#) para o ElastiCache.

Monitoramento de cluster do CloudWatch e métricas de nó (console)

Para coletar estatísticas de utilização de CPU para um cluster de cache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Selecione os nós de cache para os quais você deseja visualizar métricas.

 Note

Selecionar mais de 20 nós desabilita as métricas de visualização no console.

- a. Na página Cache Clusters (Clusters de cache) do Console de Gerenciamento da AWS, clique no nome de um ou mais clusters de cache.

A página de detalhes do cluster de cache é exibida.

- b. Clique na guia Nodes na parte superior da janela.
- c. Na guia Nodes da janela de detalhes, selecione os nós de cache para os quais você deseja visualizar as métricas.

Uma lista das Métricas do CloudWatch disponíveis aparece na parte inferior da janela do console.

- d. Clique na métrica CPU Utilization.

O console do CloudWatch será aberto, exibindo as métricas selecionadas. Você pode usar as caixas de listagem suspensas Statistic e Period e a guia Time Range para alterar as métricas exibidas.

Monitoramento de cluster do CloudWatch e métricas de nó usando a CLI do CloudWatch

Para coletar estatísticas de utilização de CPU para um cluster de cache

- Para Linux, macOS ou Unix:

```
aws cloudwatch get-metric-statistics \
  --namespace AWS/ElastiCache \
  --metric-name CPUUtilization \
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' \
  --statistics=Average \
  --start-time 2018-07-05T00:00:00 \
  --end-time 2018-07-06T00:00:00 \
```

```
--period=3600
```

Para Windows:

```
aws cloudwatch get-metric-statistics ^
  --namespace AWS/ElastiCache ^
  --metric-name CPUUtilization ^
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' ^
  --statistics=Average ^
  --start-time 2018-07-05T00:00:00 ^
  --end-time 2018-07-06T00:00:00 ^
  --period=3600
```

Monitoramento de cluster do CloudWatch e métricas de nó usando a API do CloudWatch

Para coletar estatísticas de utilização de CPU para um cluster de cache

- Chame a API do CloudWatch `GetMetricStatistics` com os seguintes parâmetros (observe que os horários de início e término são mostrados apenas como exemplos, você precisará substituir seus próprios 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>
```

Monitoramento de eventos do ElastiCache pelo Amazon SNS

Quando ocorrem eventos significativos para um cluster, o ElastiCache envia uma notificação para um tópico específico do Amazon SNS. Exemplos incluem uma falha ao adicionar um nó, êxito ao adicionar um nó, a modificação de um grupo de segurança, e outros. Ao monitorar eventos chave, você pode se manter informado sobre o atual estado dos seus clusters e, dependendo do evento, poderá executar uma ação corretiva.

Tópicos

- [Gerenciamento de notificações do Amazon SNS do ElastiCache](#)
- [Visualização de eventos do ElastiCache](#)
- [Notificações de eventos e o Amazon SNS](#)

Gerenciamento de notificações do Amazon SNS do ElastiCache

Você pode configurar o ElastiCache para enviar notificações para eventos de cluster importantes usando o Amazon Simple Notification Service (Amazon SNS). Nestes exemplos, você configurará um cluster com o nome de recurso da Amazon (ARN) de um tópico do Amazon SNS para receber notificações.

Note

Esse tópico pressupõe que você tenha se cadastrado no Amazon SNS e configurado e assinado um tópico do Amazon SNS. Para obter informações sobre como fazer isso, consulte o [Guia do desenvolvedor do Amazon Simple Notification Service](#).

Adição de um tópico do Amazon SNS

As seções a seguir mostram como adicionar um tópico do Amazon SNS usando o console da AWS, a AWS CLI ou a API do ElastiCache.

Adição de um tópico do Amazon SNS (console)

O procedimento a seguir mostra como adicionar um tópico do Amazon SNS para um cluster. Para adicionar um tópico do Amazon SNS para um grupo de replicação, na etapa 2, em vez de escolher um cluster, escolha um grupo de replicação e siga as mesmas etapas restantes.

Note

Esse processo também pode ser usado para modificar o tópico do Amazon SNS.

Para adicionar ou modificar um tópico do Amazon SNS para um cluster (console)

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Em Clusters, escolha o cluster para o qual deseja adicionar ou modificar um ARN de tópico do Amazon SNS.
3. Escolha Modify (Modificar).
4. Em Modify Cluster (Modificar cluster) em Topic for SNS Notification (Tópico para notificação do SNS), escolha o tópico SNS que você deseja adicionar ou escolha Manual ARN input (Entrada manual de ARN) e insira o ARN do tópico do Amazon SNS.
5. Escolha Modify (Modificar).

Adição de um tópico do Amazon SNS (AWS CLI)

Para adicionar ou modificar um tópico do Amazon SNS a um cluster, use o comando da AWS CLI `modify-cache-cluster`.

O seguinte exemplo de código adiciona um ARN de tópico do Amazon SNS a `my-cluster`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \
```

```
--cache-cluster-id my-cluster \  
--notification-topic-arn arn:aws:sns:us-  
west-2:123456789xxx:ElastiCacheNotifications
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Para obter mais informações, consulte [modify-cache-cluster](#).

Adição de um tópico do Amazon SNS (API do ElastiCache)

Para adicionar ou modificar um tópico do Amazon SNS para um cluster, chame a ação `ModifyCacheCluster` com os seguintes parâmetros:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=false  
&CacheClusterId=my-cluster  
&NotificationTopicArn=arn%3Aaws%3Asns%3Aus-  
west-2%3A565419523791%3AElastiCacheNotifications  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

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

Habilitação e desabilitação de notificações do Amazon SNS

Você pode ativar ou desativar notificações para um cluster. Os procedimentos a seguir mostram como desativar notificações do Amazon SNS.

Habilitação e desabilitação de notificações do Amazon SNS (console)

Para desabilitar as notificações do Amazon SNS usando o AWS Management Console

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista dos clusters que estão executando o Redis, no painel de navegação, escolha Redis.
3. Escolha a caixa à esquerda do nome do cluster para o qual você deseja modificar notificações.
4. Escolha Modify (Modificar).
5. Em Modify Cluster, em Topic for SNS Notification, escolha Disable Notifications.
6. Escolha Modify (Modificar).

Habilitação e desabilitação de notificações do Amazon SNS (AWS CLI)

Para desabilitar notificações do Amazon SNS, use o comando `modify-cache-cluster` com os seguintes parâmetros:

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-status inactive
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-status inactive
```

Habilitação e desabilitação de notificações do Amazon SNS (API do ElastiCache)

Para desabilitar notificações do Amazon SNS, chame a ação `ModifyCacheCluster` com os seguintes parâmetros:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Essa chamada retorna uma saída semelhante à seguinte:

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=false  
  &CacheClusterId=my-cluster  
  &NotificationTopicStatus=inactive  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Visualização de eventos do ElastiCache

O ElastiCache registra logs de eventos que se relacionam às suas instâncias de cluster, grupos de segurança e 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 log usando o console do ElastiCache, o comando da AWS CLI `describe-events` ou a ação `DescribeEvents` da API do ElastiCache.

Os procedimentos a seguir mostram como visualizar todos os eventos do ElastiCache das últimas 24 horas (1.440 minutos).

Visualização de eventos do ElastiCache (console)

O procedimento a seguir exibe eventos usando o console do ElastiCache.

Para visualizar eventos usando o console do ElastiCache

1. Faça login no AWS Management Console e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. Para ver uma lista de todos os eventos disponíveis, no painel de navegação, escolha Events (Eventos).

Na tela Event (Eventos), cada linha da lista representa um evento e exibe a origem desse evento, o tipo de evento (`cache-cluster`, `cache-parameter-group`, `cache-security-group` ou `cache-subnet-group`), a hora GMT do evento e uma descrição do evento.

Usando a opção `Filter`, você pode especificar se deseja ver todos os eventos ou apenas eventos de um tipo específico na lista de eventos.

Visualização de eventos do ElastiCache (AWS CLI)

Para gerar uma lista de eventos do ElastiCache usando a 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](#).

Visualização de eventos do ElastiCache (API do ElastiCache)

Para gerar uma lista de eventos do ElastiCache usando a API do ElastiCache, use a ação `DescribeEvents`. Você pode usar parâmetros opcionais para controlar os tipos de eventos listados, o período de tempo dos eventos listados, o número máximo de eventos a serem listados e muito mais.

O código a seguir lista os 40 eventos de cluster de cache mais recentes.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeEvents  
&MaxRecords=40  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&SourceType=cache-cluster  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

O código a seguir lista os eventos de cluster de cache nas últimas 24 horas (1440 minutos).

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeEvents  
&Duration=1440  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&SourceType=cache-cluster  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

As ações acima devem produzir uma saída semelhante à seguinte.

```
<DescribeEventsResponse xmlns="http://elasticache.amazonaws.com/doc/2015-02-02/">
```

```
<DescribeEventsResult>
  <Events>
    <Event>
      <Message>Cache cluster created</Message>
      <SourceType>cache-cluster</SourceType>
      <Date>2015-02-02T18:22:18.202Z</Date>
      <SourceIdentifier>mem01</SourceIdentifier>
    </Event>
    (...output omitted...)
  </Events>
</DescribeEventsResult>
<ResponseMetadata>
  <RequestId>e21c81b4-b9cd-11e3-8a16-7978bb24ffdf</RequestId>
</ResponseMetadata>
</DescribeEventsResponse>
```

Para obter mais informações, como os parâmetros disponíveis e os valores de parâmetros permitidos, consulte [DescribeEvents](#).

Notificações de eventos e o Amazon SNS

O ElastiCache pode publicar mensagens usando o Amazon Simple Notification Service (SNS) quando eventos significativos acontecerem 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 documentação do Amazon SNS, consulte a [página do produto Amazon SNS](#).

As notificações são publicadas em um tópico do Amazon SNS especificado. Os seguintes são requisitos para notificações:

- Apenas um tópico pode ser configurado para notificações do ElastiCache.
- A conta da AWS que possui o tópico do Amazon SNS deve ser a mesma conta que possui o cluster de cache em que as notificações estão habilitadas.

- O tópico do Amazon SNS no qual você está publicando não pode ser criptografado.

Note

É possível anexar um tópico criptografado (em repouso) do Amazon SNS ao cluster. No entanto, o status do tópico do console do ElastiCache será exibido como inativo, o que efetivamente disassocia o tópico do cluster quando o ElastiCache envia mensagens para o tópico.

- O tópico do Amazon SNS deve estar na mesma região que o cluster do ElastiCache.

Eventos do ElastiCache

Os seguintes eventos do ElastiCache acionam notificações do Amazon SNS. Para obter informações sobre detalhes de eventos, consulte [Visualização de eventos do ElastiCache](#).

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 a endereços IP livres insuficientes	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.

Nome do evento	Message	Descrição
ElastiCache:CacheClusterProvisioningFailed devido a estado de rede incompatível	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Foi feita uma tentativa de executar um novo cluster de cache em uma nuvem privada virtual (VPC) inexistente.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Escalabilidade para cluster de cache concluída com êxito.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	A operação de expansão no cluster de cache falhou.
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	Um dos seguintes eventos ocorreu: <ul style="list-style-type: none"> A lista de security groups de cache autorizados para o cluster de cache foi modificada. Um ou mais novos security groups do EC2 foram autorizados em qualquer um dos security groups de cache associados ao cluster de cache. Um ou mais security groups do EC2 foram revogados de qualquer um dos security groups de cache associados ao cluster de cache.

Nome do evento	Message	Descrição
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>O ElastiCache detectou que o host que está executando um nó de cache está degradado ou é inacessível e iniciou a substituição do nó de cache.</p> <div data-bbox="1068 495 1510 810"><p> Note</p><p>A entrada de DNS para o nó de cache substituído não é 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 de clientes de cache podem parar de usar o nó de cache mesmo após o ElastiCache ter substituído o nó de cache. Nesse caso, a aplicação deve atualizar a lista de servidores quando esse evento ocorrer.</p>

Nome do evento	Message	Descrição
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>O ElastiCache detectou que o host que executa um nó de cache está degradado ou é inacessível e concluiu a substituição do nó de cache.</p> <div data-bbox="1068 495 1507 806"><p> Note</p><p>A entrada de DNS para o nó de cache substituído não é 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 de clientes de cache podem parar de usar o nó de cache mesmo após o ElastiCache ter substituído o nó de cache. Nesse caso, a aplicação deve atualizar a lista de servidores quando esse evento ocorrer.</p>

Nome do evento	Message	Descrição
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Um ou mais nós de cache foram reinicializados. Mensagem (Memcached): "Cache node %s shutdown" Em seguida, uma segunda mensagem: "Cache node %s restarted"
ElastiCache:CertificateRenewalComplete (somente Redis)	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 Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	O failover para um nó de réplica foi bem-sucedido.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	O número de réplicas no cluster foi aumentado.

Nome do evento	Message	Descrição
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	O processo de adição de réplicas ao cluster começou.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Um nó no seu cluster que estava programado para substituição já não está programado para substituição.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Um nó no seu cluster previamente programado para substituição foi reprogramado para substituição durante a nova janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós .
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Um nó no seu cluster está programado para substituição durante a janela descrita na notificação. Para obter informações sobre quais ações você pode realizar, consulte Substituição de nós .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Um nó de cache foi removido do cluster de cache.

Nome do evento	Message	Descrição
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	A operação de expansão no grupo de replicação foi concluída com sucesso.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	A operação de expansão no grupo de replicação falhou.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Uma atualização de autoatendimento está disponível para o nó.
ElastiCache:SnapshotComplete (somente Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Um snapshot de cache foi concluído com sucesso.
ElastiCache:SnapshotFailed (somente Redis)	SnapshotFailed : <i>cluster-name</i>	Um snapshot de cache falhou. Consulte os eventos de cache do cluster para mais uma causa detalhada. Se você descrever o snapshot, consulte DescribeSnapshots , o status será failed.

Tópicos relacionados

- [Visualização de eventos do ElastiCache](#)

Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail

O Amazon ElastiCache é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, por uma função ou por um serviço da AWS no Amazon ElastiCache. O CloudTrail captura todas as chamadas de API para o Amazon ElastiCache como eventos, inclusive as chamadas do console do Amazon ElastiCache e de chamadas do código para operações da API do Amazon ElastiCache. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Amazon ElastiCache. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o Amazon ElastiCache, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail CloudTrail](#).

Informações sobre o Amazon ElastiCache no CloudTrail

O CloudTrail é habilitado em sua conta da AWS quando ela é criada. Quando ocorre uma atividade no Amazon ElastiCache, essa atividade é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS no Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte [Como visualizar eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro de eventos em andamento na sua conta da AWS, incluindo eventos do Amazon ElastiCache, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configurar notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#) e [receber arquivos de log do CloudTrail de várias contas](#)

Todas as ações do Amazon ElastiCache são registradas pelo CloudTrail e estão documentadas na [Referência da API do ElastiCache](#). Por exemplo, as chamadas para as ações `CreateCacheCluster`, `DescribeCacheCluster` e `ModifyCacheCluster` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte o [Elemento `userIdentity` do CloudTrail](#).

Noções básicas sobre entradas de arquivo de log do Amazon ElastiCache

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateCacheCluster`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "eventTime": "2014-12-01T22:00:35Z",
  "eventSource": "elasticache.amazonaws.com",
  "eventName": "CreateCacheCluster",
```

```

"awsRegion":"us-west-2",
"sourceIPAddress":"192.0.2.01",
"userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
"requestParameters":{
  "numCacheNodes":2,
  "cacheClusterId":"test-memcached",
  "engine":"memcached",
  "aZMode":"cross-az",
  "cacheNodeType":"cache.m1.small",
},
"responseElements":{
  "engine":"memcached",
  "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/home#client-download:",
  "cacheParameterGroup":{
    "cacheParameterGroupName":"default.memcached1.4",
    "cacheNodeIdsToReboot":{
    },
    "parameterApplyStatus":"in-sync"
  },
  "preferredAvailabilityZone":"Multiple",
  "numCacheNodes":2,
  "cacheNodeType":"cache.m1.small",

  "cacheClusterStatus":"creating",
  "autoMinorVersionUpgrade":true,
  "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
  "cacheClusterId":"test-memcached",
  "engineVersion":"1.4.14",
  "cacheSecurityGroups":[
    {
      "status":"active",
      "cacheSecurityGroupName":"default"
    }
  ],
  "pendingModifiedValues":{
  }
},
"requestID":"104f30b3-3548-11e4-b7b8-6d79ffe84edd",
"eventID":"92762127-7a68-42ce-8787-927d2174cde1"
}

```

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `DescribeCacheCluster`. Observe que, para todas as chamadas `Describe` do Amazon ElastiCache (`Describe*`), a seção `ResponseElements` é removida e aparece como `null`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "eventTime": "2014-12-01T22:01:00Z",
  "eventSource": "elasticache.amazonaws.com",
  "eventName": "DescribeCacheClusters",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.01",
  "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters": {
    "showCacheNodeInfo": false,
    "maxRecords": 100
  },
  "responseElements": null,
  "requestID": "1f0b5031-3548-11e4-9376-c1d979ba565a",
  "eventID": "a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

O exemplo a seguir mostra uma entrada de log do CloudTrail que registra uma ação `ModifyCacheCluster`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "responseElements": null,
  "requestID": "1f0b5031-3548-11e4-9376-c1d979ba565a",
  "eventID": "a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

```
"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"  
}
```

Cotas do ElastiCache

Sua conta da AWS possui cotas padrão, anteriormente chamadas de limites, para cada produto da AWS. A menos que especificado de outra forma, cada cota é específica da região. Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Para visualizar as cotas do ElastiCache, abra o [console Service Quotas](#) (Cotas de serviço). No painel de navegação, selecione serviços da AWS e ElastiCache.

Para solicitar o aumento da cota, consulte [Requesting a Quota Increase](#) (Solicitação de aumento de cota) no Guia do usuário do Service Quotas. Se a cota ainda não estiver disponível no Service Quotas, use o [formulário de aumento de limite](#).

Sua conta da AWS tem as seguintes cotas relacionadas ao ElastiCache.

Recurso	Padrão
Caches sem servidor por região	40
Instantâneos sem servidor por dia por cache	24
Nós por região	300
Nós por cluster por tipo de instância (modo cluster do Redis habilitado)	90
Fragmentos por cluster (modo cluster do Redis desabilitado)	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-rede	20
Usuários por grupo de usuários	100
Número máximo de usuários	1000

Recurso	Padrão
Número máximo de grupos de usuários	100

Referência

Os tópicos desta seção abrangem o trabalho com a API do Amazon ElastiCache e a seção do ElastiCache da AWS CLI. Também estão incluídas nesta seção mensagens de erro comuns e notificações de serviço.

- [Uso da API do ElastiCache](#)
- [Referência da API do ElastiCache](#)
- [Seção ElastiCache da Referência da AWS CLI](#)
- [Mensagens de erro do Amazon ElastiCache](#)
- [Notificações](#)

Uso da API do ElastiCache

Esta seção fornece descrições orientadas a tarefas sobre como usar e implementar operações do ElastiCache. Para uma descrição completa dessas operações, consulte a [Referência de API do Amazon ElastiCache](#)

Tópicos

- [Como usar a API de consulta](#)
- [Bibliotecas disponíveis](#)
- [Solução de problemas de aplicações](#)

Como usar a API de consulta

Parâmetros de consulta

As solicitações baseadas em consulta HTTP são solicitações HTTP que usam o verbo HTTP GET ou POST e um parâmetro de consulta chamado `Action`.

Cada solicitação de consulta deve incluir alguns parâmetros comuns para lidar com a autenticação e a seleção de uma ação.

Algumas operações levam listas de parâmetros. Essas listas são especificadas usando a notação `param.n`. Os valores de `n` são inteiros a partir de 1.

Autenticação de solicitação de consulta

Só é possível enviar solicitações de consulta por meio de HTTPS, e é preciso incluir uma assinatura em todas as solicitações de consulta. Esta seção descreve como criar a assinatura. O método descrito no procedimento a seguir é conhecido como versão de assinatura 4.

As etapas básicas a seguir são usadas para autenticar as solicitações à AWS. Isso presume que você esteja registrado na AWS e possua um ID de chave de acesso e uma chave de acesso secreta.

Processo de autenticação de consulta

1. O remetente elabora uma solicitação à AWS.
2. O remetente calcula a assinatura da solicitação, um hash codificado para o HMAC (Hash-based Message Authentication Code) com uma função de hash SHA-1, conforme definido na próxima seção deste tópico.
3. O remetente da solicitação envia os dados da solicitação, a assinatura e o ID da chave de acesso (o identificador da chave de acesso secreta usada) à AWS.
4. A AWS usa o ID de chave de acesso para pesquisar pela chave de acesso secreta.
5. A 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 enviada na solicitação.
6. Se as assinaturas coincidirem, a solicitação será considerada autêntica. Se a comparação falhar, a solicitação será descartada e a AWS retornará uma resposta de erro.

Note

Se uma solicitação contiver um parâmetro `Timestamp`, a assinatura calculada para a solicitação expirará 15 minutos após o valor.

Se uma solicitação contiver um parâmetro `Expires`, a assinatura expirará no horário especificado pelo parâmetro `Expires`.

Para calcular a assinatura da solicitação

1. Crie a query string canonizada de que você precisará posteriormente neste procedimento:

- a. Classifique os componentes query string UTF-8 por nome do parâmetro com o ordenamento natural de bytes. Os parâmetros podem vir do URI GET ou do corpo POST (quando Content-Type for application/x-www-form-urlencoded).
 - b. Codificar em URL o nome do parâmetro e os valores de acordo com as seguintes regras:
 - i. Não codificar em URL nenhum dos caracteres não reservados que definem o RFC 3986. Esses caracteres não reservados são A–Z, a–z, 0–9, hífen (-), sublinhado (_), ponto (.) e til (~).
 - ii. Codificar em percentual todos os outros caracteres com %XY, onde X e Y são caracteres hexadecimais de 0 a 9 e maiúsculas de A a F.
 - iii. Codificar em percentual os caracteres UTF-8 estendidos na forma %XY%ZA...
 - iv. Codificar em percentual o caractere de espaço como %20 (e não +, como em esquemas de codificação comuns).
 - c. Separe os nomes de parâmetro codificados a partir de seus valores codificados com o sinal de igual (=) (caractere ASCII 61), mesmo se o valor do parâmetro estiver vazio.
 - d. Separe os pares de nome-valor por um "&" (e comercial) (código 38 em ASCII).
2. Crie a string para assinar de acordo com a seguinte pseudogramática ("\\n" representa uma nova linha em ASCII).

```
StringToSign = HTTPVerb + "\\n" +  
ValueOfHostHeaderInLowercase + "\\n" +  
HTTPRequestURI + "\\n" +  
CanonicalizedQueryString <from the preceding step>
```

O componente HTTPRequestURI é o componente de caminho absoluto HTTP do URI até a query string (não incluída). Se o HTTPRequestURI estiver vazio, use uma barra (/).

3. Calcule um HMAC compatível com RFC 2104 com a string recém-criada, sua chave de acesso secreta como chave e SHA256 ou SHA1 como algoritmo de hash.

Para obter mais informações, consulte <https://www.ietf.org/rfc/rfc2104.txt>.

4. Converta o valor resultante para base64.
5. Inclua o valor como o valor do parâmetro Signature na solicitação.

Por exemplo, a seguir você encontra um exemplo de solicitação (as quebras de linha foram adicionadas para maior clareza).

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01
```

Quanto à string de consulta anterior, você calcularia a assinatura HMAC na seguinte string.

```
GET\n  
  elasticache.amazonaws.com\n  Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache  
%2Faws4_request  
  &X-Amz-Date=20141201T223649Z  
  &X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-  
amz-date  
    content-type:  
    host:elasticache.us-west-2.amazonaws.com  
    user-agent:CacheServicesAPICommand_Client  
  x-amz-content-sha256:  
  x-amz-date:
```

O resultado é a seguinte solicitação assinada.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterIdentifier=myCacheCluster  
  &SignatureMethod=HmacSHA256  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request
```

```
&X-Amz-Date=20141201T223649Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

Para obter informações detalhadas sobre o processo de assinatura e o cálculo da assinatura da solicitação, consulte o tópico [Processo de assinatura do Signature Version 4](#) e seus subtópicos.

Bibliotecas disponíveis

A AWS fornece kits de desenvolvimento de software (SDKs) para desenvolvedores de software que preferem criar aplicações usando APIs específicas de linguagem em vez da API de consulta. Esses SDKs oferecem as funções básicas (não incluídas nas APIs), como autenticação de solicitação, novas tentativas de solicitação e processamento de erros, para que você possa começar a usar com mais facilidade. SDKs e recursos adicionais estão disponíveis para as seguintes linguagens de programação:

- [Java](#)
- [Windows and .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Para obter informações sobre outras linguagens, consulte [Código e bibliotecas de exemplo](#).

Solução de problemas de aplicações

O ElastiCache fornece erros específicos e descritivos para ajudar você a solucionar problemas enquanto interage com a API do ElastiCache.

Recuperação de erros

Normalmente, espera-se que o aplicativo verifique se uma solicitação gerou um erro antes que você precise processar os resultados. A maneira mais fácil de descobrir se ocorreu um erro é procurar por um nó de `Error` na resposta da API do ElastiCache.

A sintaxe XPath apresenta uma maneira simples de procurar pela presença de um nó `Error`, bem como uma maneira fácil de recuperar o código e a mensagem de erro. O snippet de código a seguir

usa Perl e o módulo XML::XPath para determinar se ocorreu um erro durante uma solicitação. Caso tenha ocorrido, o código imprimirá o primeiro código de erro e a mensagem na resposta.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Dicas de solução de problemas

Recomendamos os seguintes processos para diagnosticar e resolver problemas com a API do ElastiCache.

- Verifique se o ElastiCache está funcionando corretamente.

Para fazer isso, basta abrir uma janela do navegador e enviar uma solicitação de consulta para o serviço do ElastiCache (como <https://elasticsearch.amazonaws.com>). Uma `MissingAuthenticationTokenException` ou um Erro do servidor interno 500 confirma que o serviço está disponível e respondendo a solicitações.

- Verificação da estrutura de sua solicitação.

Cada operação do ElastiCache tem uma página de referência na Referência da API do ElastiCache. Verifique novamente se você está usando os parâmetros corretamente. Para conceder ideias sobre o que pode estar errado, consulte as amostras de solicitações ou cenários de usuários para ver se esses exemplos estão realizando operações similares.

- Verificação do fórum.

O ElastiCache tem um fórum de discussão onde você pode procurar soluções para os problemas que outros usuários enfrentaram ao longo do caminho. Para exibir o fórum, consulte

<https://forums.aws.amazon.com/> .

Configuração da interface de linha de comando do ElastiCache

Esta seção descreve os pré-requisitos para executar as ferramentas da linha de comando, em que lugar obtê-las e como configurar as ferramentas e o ambiente, e inclui ainda uma série de exemplos comuns de uso das ferramentas.

Siga as instruções neste tópico somente se for usar a AWS CLI para o ElastiCache.

Important

A CLI (Command Line Interface) do Amazon ElastiCache não oferece suporte a melhorias do ElastiCache após a versão 2014-09-30 da API. Para usar a funcionalidade do ElastiCache mais recente na linha de comando, use a opção [Interface de linha de comando da AWS](#).

Tópicos

- [Pré-requisitos](#)
- [Obtenção das ferramentas da linha de comando](#)
- [Configuração das ferramentas](#)
- [Como fornecer credenciais para as ferramentas](#)
- [Variáveis de ambiente](#)

Pré-requisitos

Este documento pressupõe que você pode trabalhar em um ambiente Linux/UNIX ou Windows. As ferramentas da linha de comando do Amazon ElastiCache também funcionam no Mac OS X, que é um ambiente baseado no UNIX; contudo, não há nenhuma instrução específica ao Mac OS X neste guia.

Como convenção, todos os textos da linha de comando recebem prefixo com um prompt da linha de comando genérico **PROMPT>** . O prompt da linha de comando real da sua máquina pode ser diferente. Também usamos **\$** para indicar um comando específico do Linux/UNIX e **C:\>** para um comando específico do Windows. O exemplo de saída resultante do comando é mostrado imediatamente após, sem qualquer prefixo.

O Java Runtime Environment

As ferramentas da linha de comando usadas neste guia requerem a execução do Java versão 5 ou posterior. Uma instalação do JRE ou do JDK é aceitável. Para ver e fazer download de JREs para uma série de plataformas, incluindo Linux/UNIX e Windows, consulte [Java SE Downloads](#).

Configuração da variável home do Java

As ferramentas da linha de comando dependem de uma variável de ambiente (JAVA_HOME) para localizar o Java Runtime. Essa variável de ambiente deve ser definida para o caminho completo do diretório que contém um subdiretório denominado bin, que, por sua vez, contém o executável java (no Linux e no UNIX) ou o executável java.exe (no Windows).

Para configurar da variável Java Home

1. Configure a variável Java Home.

- No Linux e UNIX, digite o comando a seguir:

```
$ export JAVA_HOME=<PATH>
```

- No Windows, digite o seguinte comando:

```
C:\> set JAVA_HOME=<PATH>
```

2. Confirme a configuração do caminho executando \$JAVA_HOME/bin/java -version e verificando a saída.

- No Linux/UNIX, você verá uma saída semelhante à seguinte:

```
$ $JAVA_HOME/bin/java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

- No Windows, você verá uma saída semelhante à seguinte:

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

Obtenção das ferramentas da linha de comando

As ferramentas da linha de comando estão disponíveis como arquivo ZIP no [site Ferramentas de desenvolvedor do ElastiCache](#). Essas ferramentas são escritas em Java e incluem scripts de shell para Windows 2000/XP/Vista/Windows 7, Linux/UNIX e Mac OSX. O arquivo ZIP é independente e nenhuma instalação é necessária; basta fazer download do arquivo ZIP e decompactá-lo em um diretório de seu computador local.

Configuração das ferramentas

As ferramentas da linha de comando dependem de uma variável de ambiente (AWS_ELASTICACHE_HOME) para localizar bibliotecas compatíveis. Você precisa definir essa variável de ambiente para poder usar as ferramentas. Indique o caminho do diretório no qual você descompactou as ferramentas da linha de comando. Esse diretório é chamado ElastiCacheCli-A.B.nnnn (A, B e n são os números de versão/release) e contém os subdiretórios denominados bin e lib.

Para definir a variável de ambiente AWS_ELASTICACHE_HOME

- Abra uma janela de linha de comando e insira um dos seguintes comandos para definir a variável de ambiente AWS_ELASTICACHE_HOME.
 - No Linux e UNIX, digite o comando a seguir:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- No Windows, digite o seguinte comando:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Para facilitar o uso das ferramentas, é recomendável adicionar o diretório BIN das ferramentas ao CAMINHO do sistema. O restante deste guia pressupõe que o diretório BIN encontra-se no caminho do sistema.

Para adicionar o diretório BIN das ferramentas ao caminho do sistema

- Insira os comandos a seguir para adicionar o diretório BIN das ferramentas ao CAMINHO do sistema.

- No Linux e UNIX, digite o comando a seguir:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- No Windows, digite o seguinte comando:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

Note

As variáveis do ambiente Windows são redefinidas quando você fecha a janela de comando. Talvez você queira defini-las permanentemente. Consulte a documentação da versão de seu Windows para obter mais informações.

Note

Os caminhos que contêm um espaço devem ser colocados entre aspas duplas; por exemplo: “C:\Program Files\Java”

Como fornecer credenciais para as ferramentas

As ferramentas da linha de comando precisam da chave de acesso e da chave de acesso secreta da AWS fornecidas com sua conta da AWS. Você pode obtê-las usando a linha de comando ou um arquivo de credenciais localizado no sistema local.

A implantação inclui um arquivo de modelo `${AWS_ELASTICACHE_HOME}/credential-file-path.template` que você precisa alterar com suas informações. A seguir encontra-se o conteúdo do arquivo de modelo:

```
AWSAccessKeyId=<Write your AWS access ID>  
AWSSecretKey=<Write your AWS secret key>
```

⚠ Important

No UNIX, limite as permissões ao proprietário do arquivo de credenciais:

```
$ chmod 600 <the file created above>
```

Com o arquivo de credenciais configurado, você precisará definir a variável de ambiente `AWS_CREDENTIAL_FILE` para que as ferramentas do ElastiCache possam encontrar suas informações.

Para definir a variável de ambiente `AWS_CREDENTIAL_FILE`

1. Defina a variável de ambiente:

- No Linux e UNIX, atualize a variável usando o seguinte comando:

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- No Windows, defina a variável usando o seguinte comando:

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. Verifique se sua configuração está funcionando corretamente com o seguinte comando:

```
elasticache --help
```

Você deve examinar a página de uso de todos os comandos do ElastiCache.

Variáveis de ambiente

As variáveis de ambiente podem ser úteis para scripting, configuração de padrões ou para substituí-los temporariamente.

Além da variável de ambiente `AWS_CREDENTIAL_FILE`, a maioria das ferramentas de API incluídas com a interface da linha de comando do ElastiCache comporta as seguintes variáveis:

- `EC2_REGION`: a região da AWS a ser usada.

- `AWS_ELASTICACHE_URL`: o URL a ser usado para a chamada de serviço. Não é necessária para especificar um endpoint regional diferente quando `EC2_REGION` for especificada ou quando se passa o parâmetro `--region`.

Os exemplos a seguir mostram como se define a variável de ambiente `EC2_REGION` para configurar a região usada pelas ferramentas de API:

Linux, OSX ou UNIX

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

Mensagens de erro do Amazon ElastiCache

As seguintes mensagens de erro são retornadas pelo Amazon ElastiCache. Você pode receber outras mensagens de erro retornadas pelo ElastiCache, por outros serviços da AWS ou pelo Redis. Para obter descrições de mensagens de erro de fontes diferentes do 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 de nó do Amazon ElastiCache](#).

Para ter mais informações, consulte [Limites do Amazon ElastiCache](#) no 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 de nó do Amazon ElastiCache](#).

Para ter mais informações, consulte [Limites do Amazon ElastiCache](#) no 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 para Redis](#).

Notificações

Este tópico discute as notificações do ElastiCache nas quais você possa estar interessado. Uma notificação é uma situação ou evento que, na maioria dos casos, é temporário, durando apenas até que uma solução seja encontrada e implementada. Em geral, as notificações têm uma data de início e uma data de resolução, após a qual elas deixam de ser relevantes. Qualquer notificação pode ou não ser relevante para você. Recomendamos uma diretriz de implementação que, se seguida, vai melhorar o desempenho do seu cluster.

Notificações não anunciam recursos ou funcionalidades novas ou melhoradas do ElastiCache.

Notificações gerais do ElastiCache

Atualmente não há nenhuma notificação importante do ElastiCache que não seja específica ao mecanismo.

Notificações específicas do ElastiCache for Redis

Atualmente não há nenhuma notificação importante do ElastiCache for Redis.

ElastiCache para o histórico da documentação do Redis

- Versão da API: 02/02/2015
- Última atualização da documentação: 27 de novembro de 2023

A tabela a seguir descreve mudanças importantes em cada versão do para Redis após março de 2018. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em o feed RSS.

Atualizações ElastiCache recentes do Redis

Alteração	Descrição	Data
ElastiCache para Redis foi adicionado suporte para tamanhos adicionais de nós C7gn	ElastiCache para Redis adicionou suporte para tamanhos adicionais de nós C7gn.	10 de janeiro de 2024
ElastiCache for Redis agora suporta a criação de caches sem servidor	Agora você pode criar caches sem servidor, que simplificam o gerenciamento de cache e escalam instantaneamente para dar suporte a aplicações mais exigentes. Para obter mais informações, consulte Choosing between deployment options . Como parte desse recurso, novas permissões foram adicionadas a <code>ElastiCacheServiceRolePolicy</code> e a <code>AmazonElastiCacheFullAccess</code> para permitir a associação de caches sem servidor com endpoints da VPC gerenciados. Além	27 de novembro de 2023

disso, as permissões foram adicionadas para dar suporte a uma experiência revisada do console usando a política AmazonElastiCacheFullAccess .

[ElastiCache para Redis 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 for Redis agora suporta a modificação das configurações de criptografia em trânsito](#)

Agora você pode alterar a configuração de TLS dos seus clusters do Redis sem a necessidade de criar ou provisionar de novo os clusters ou afetar a disponibilidade da aplicação. Para obter mais informações, consulte [Habilitação da criptografia em trânsito em um cluster existente](#).

28 de dezembro de 2022

[ElastiCache for Redis agora oferece suporte à autenticação de usuários usando o IAM](#)

A autenticação do IAM permite que você autentique uma conexão com o ElastiCache Redis usando identidades AWS do IAM. Isso possibilita que você fortaleça seu modelo de segurança e simplifique várias tarefas administrativas de segurança. Para obter mais informações, consulte [Autenticação com o IAM](#).

16 de novembro de 2022

[ElastiCache para Redis agora suporta Redis 7](#)

Esta versão traz vários novos recursos para a Amazon ElastiCache for Redis: funções do Redis, melhorias na ACL e Sharded Pub/Sub. Para obter mais informações, consulte ElastiCache a [versão 7.0 do Redis](#).

8 de novembro de 2022

[ElastiCache para Redis agora suporta IPV6](#)

ElastiCache suporta as versões 4 e 6 do Protocolo de Internet (IPv4 e IPv6), permitindo que você configure seu cluster para aceitar somente conexões IPv4, somente conexões IPv6 ou conexões IPv4 e IPv6 (pilha dupla). O IPv6 é compatível com workloads usando o mecanismo Redis versão 6.2 em diante em todas as instâncias criadas no [sistema Nitro](#). Não há cobranças adicionais pelo acesso ElastiCache via IPv6. Para obter mais informações, consulte [Escolher um tipo de rede](#).

7 de novembro de 2022

[ElastiCache for Redis agora oferece suporte ao formato nativo de notação de JavaScript objeto \(JSON\)](#)

O formato nativo de notação de JavaScript objetos (JSON) é uma forma simples e sem esquemas de codificar conjuntos de dados complexos dentro de clusters do Redis. Você pode armazenar e acessar dados de forma nativa usando o formato JavaScript Object Notation (JSON) dentro dos clusters do Redis e atualizar os dados JSON armazenados nesses clusters, sem precisar gerenciar código personalizado para serializá-los e desserializá-los. Para obter mais informações, consulte [Conceitos básicos do JSON](#).

25 de maio de 2022

[ElastiCache agora suporta PrivateLink](#)

AWS PrivateLink permite que você acesse de forma privada as operações ElastiCache da API sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão AWS Direct Connect. Para obter mais informações, consulte [Amazon ElastiCache API e interface VPC endpoints \(AWS PrivateLink\)](#) para Redis ou [ElastiCache Amazon API e interface VPC endpoints](#) () para Memcached. AWS PrivateLink

24 de janeiro de 2022

[ElastiCache para Redis agora oferece suporte ao Redis 6.2 e à hierarquização de dados](#)

O Amazon ElastiCache for Redis apresenta a próxima versão do mecanismo Redis suportado pela Amazon. ElastiCache para Redis 6.2 inclui melhorias de desempenho para clusters habilitados para TLS usando tipos de nós x86 com 8 vCPUs ou mais ou tipos de nós Graviton2 com 4 vCPUs ou mais. ElastiCache for Redis também introduz a hierarquização 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 a [versão 6.2 \(aprimorada\) do Redis](#) e a [classificação por camadas de dados](#).

23 de novembro de 2021

[Suporte para o Auto Scaling](#)

ElastiCache for Redis agora oferece suporte ao Auto Scaling. ElastiCache para Redis, o auto scaling é a capacidade de aumentar ou diminuir automaticamente os fragmentos ou réplicas desejados em seu serviço ElastiCache for Redis. ElastiCache aproveita o serviço Application Auto Scaling para fornecer essa funcionalidade. Para obter mais informações, consulte [Auto Scaling ElastiCache para clusters Redis](#).

19 de agosto de 2021

[Suporte para a entrega de logs lentos do Redis](#)

ElastiCache agora permite que você transmita o Redis SLOWLOG para um dos dois destinos: Amazon Data Firehose ou Amazon Logs. CloudWatch Para obter mais informações, consulte [Entrega de log](#).

22 de abril de 2021

[Suporte à marcação de recursos e chaves de condição](#)

ElastiCache agora oferece suporte à marcação para ajudar você a gerenciar seus clusters e outros ElastiCache recursos. Para obter mais informações, consulte [Como marcar seus ElastiCache recursos](#). ElastiCache também introduz suporte para chaves de condição. Você pode especificar as condições que determinam como uma política do IAM entra em vigor. Para obter mais informações, consulte [Uso de chaves de condição](#).

7 de abril de 2021

[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 [Uso de Outposts](#) para Redis, ou [Uso de Outposts](#) para Memcached.

8 de outubro de 2020

[ElastiCache agora suporta Redis 6](#)

O Amazon ElastiCache for Redis apresenta a próxima versão do mecanismo Redis suportado pela Amazon. ElastiCache Esta versão inclui [Autenticação de usuários com controle de acesso baseado em função](#), suporte sem versão, cache do lado do cliente e melhorias operacionais significativas. Para obter mais informações, consulte ElastiCache a [versão 6.0 \(aprimorada\) do Redis](#).

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 for Redis agora suporta a escalabilidade de seu ambiente Redis Cluster para até 500 nós ou 500 fragmentos](#)

O modo cluster do Redis possibilita configurações que você pode usar para particionar seus dados em vários fragmentos e oferece melhor escalabilidade, performance e disponibilidade. Esse recurso está disponível no Amazon ElastiCache for Redis versão 5.0.6 em diante em todas as AWS regiões e para todos os ambientes existentes e novos do Redis ElastiCache Cluster. Para obter mais informações, consulte [Nós e fragmentos do Redis](#).

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 for Redis adiciona métricas adicionais da Amazon CloudWatch](#)

ElastiCache for Redis agora oferece suporte a novas CloudWatch métricas, incluindo PubSubCmds e HyperLogLogBasedCmds. Para obter uma lista completa, consulte [Métricas para Redis](#).

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 for Redis agora oferece suporte ao Global Datastore for Redis](#)

O 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 o Redis ElastiCache para permitir leituras de baixa latência e recuperação de desastres em todas as regiões. AWS Você pode criar, modificar e descrever um datastore global. Você também pode adicionar ou remover AWS regiões do seu armazenamento de dados global e promover uma AWS região como primária dentro de um armazenamento de dados global. Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global](#).

16 de março de 2020

[ElastiCache para Redis agora suporta a versão 5.0.6 do Redis](#)

Para obter mais informações, consulte ElastiCache a [versão 5.0.6 \(aprimorada\) do Redis](#).

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 do Amazon EC2 oferecem um nível de linha de base de desempenho de CPU, além da capacidade de intermitência do uso de CPU a qualquer momento até o consumo dos créditos acumulados. Para obter mais informações, consulte [Tipos de nó com suporte](#).

12 de novembro de 2019

[A Amazon ElastiCache agora suporta a modificação do token AUTH em um servidor existente ElastiCache para Redis](#)

ElastiCache para Redis 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 AUTH](#).

30 de outubro de 2019

[A Amazon ElastiCache agora oferece suporte à migração de dados on-line do Redis no Amazon EC2](#)

Agora você pode usar o Online Migration para migrar seus dados do Redis auto-hospedado no Amazon EC2 para a Amazon. ElastiCache Para obter mais informações, consulte [Migração online para ElastiCache](#).

28 de outubro de 2019

[ElastiCache for Redis introduz o escalonamento vertical on-line para o modo Redis Cluster.](#)

Agora você pode ampliar ou reduzir seu cluster Redis fragmentado sob demanda. ElastiCache for Redis 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 para Redis agora permite que os usuários usem um único endpoint de leitura para seu cluster Amazon ElastiCache for Redis.](#)

Esse recurso permite que você direcione todo o tráfego de leitura ElastiCache para seu cluster for Redis 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 for Redis agora permite que os usuários apliquem atualizações de serviço de acordo com sua própria programação](#)

Com este recurso, você pode optar por aplicar atualizações de serviços disponíveis no horário de sua escolha, e não apenas durante as janelas de manutenção. Isso minimizará as interrupções do serviço, especialmente durante os picos de fluxo de negócios, e ajudará a garantir que você permaneça em conformidade se seu cluster estiver em programas ElastiCache de conformidade compatíveis. Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache](#) e [Validação de conformidade para a Amazon ElastiCache](#).

4 de junho de 2019

[ElastiCache Ofertas de instância reservada padrão: adiantamento parcial, pagamento adiantado total e sem pagamento adiantado.](#)

As instâncias reservadas oferecem a flexibilidade de reservar uma ElastiCache instância da Amazon por um período de um ou três anos com base em um tipo de instância e AWS região. Para obter mais informações, consulte [Gerenciamento de custos com nós reservados](#).

18 de janeiro de 2019

[ElastiCache para suporte do Redis para até 250 nós por cluster do Redis](#)

O limite de nós ou fragmentos pode ser aumentado para um máximo de 250 ElastiCache por cluster Redis. Para obter mais informações, consulte [Estilhaços](#).

19 de novembro de 2018

[ElastiCache suporte para Redis para failover automático, backup e restauração em todos os nós T2](#)

ElastiCache for Redis introduz suporte para failover automático, criação de instantâneos e backup e restauração em todos os nós T2. Para obter mais informações, consulte [ElastiCache Backup and Restore e Snapshot do Redis](#).

19 de novembro de 2018

[ElastiCache para suporte do Redis para nós M5 e R5](#)

ElastiCache para Redis 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 for Redis 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 ElastiCache usuário do Redis. Veja também [DecreaseReplicaCount](#) [IncreaseReplicaCount](#) na Referência da ElastiCache API.

17 de setembro de 2018

[Certificação de compatibilidade FedRAMP](#)

ElastiCache for Redis agora está certificado em conformidade com o FedRAMP. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

30 de agosto de 2018

[Atualizações do mecanismo do Redis \(modo cluster habilitado\)](#)

O Amazon ElastiCache for Redis adicionou suporte para atualizar as versões do mecanismo Redis (modo de cluster ativado). Para obter mais informações, consulte [Atualizar versões do mecanismo](#).

20 de agosto de 2018

[Certificação de conformidade com PCI DSS](#)

ElastiCache for Redis agora é certificado para conformidade com PCI DSS. Para obter mais informações, consulte [Validação de conformidade para a Amazon ElastiCache](#).

5 de julho de 2018

[Support ElastiCache for Redis 4.0.10](#)

ElastiCache para Redis agora oferece suporte ao Redis 4.0.10, incluindo criptografia e redimensionamento de clusters on-line em uma única versão. Para obter mais informações, consulte ElastiCache a [versão 4.0.10 \(aprimorada\) do Redis](#).

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 \(para o Guia do Usuário do Redis\) e ElastiCache para o Memcached \(ElastiCache para o Guia do Usuário do Memcached\)](#). A estrutura da documentação na seção [AWS CLI Command Reference: elasticache](#) e na [Amazon ElastiCache API Reference permanece inalterada](#).

20 de abril de 2018

[Suporte para métrica EngineCPUUtilization](#)

ElastiCache for Redis adicionou uma nova métrica, `EngineCPUUtilization`, que relata a porcentagem da capacidade e da sua CPU que está sendo usada atualmente. Para obter mais informações, consulte [Métricas do Redis](#).

9 de abril de 2018

A tabela a seguir descreve as mudanças importantes no para Redis antes de março de 2018.

Alteração	Descrição	Alterado em
Suporte para a região Ásia-Pacífico (Osaka-local).	<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:</p> <ul style="list-style-type: none"> • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis 	12 de fevereiro de 2018
Suporte para a Europa (Paris).	<p>ElastiCache adicionou suporte para a região da UE (Paris). Para obter mais informações, consulte:</p> <ul style="list-style-type: none"> • Supported Regions (Regiões compatíveis) • Tipos de nó de cache compatíveis 	18 de dezembro de 2017
Suporte para a região da China (Ningxia)	<p>A Amazon ElastiCache adicionou suporte para a região da China (Ningxia). Para obter mais informações, consulte:</p> <ul style="list-style-type: none"> • 	11 de dezembro de 2017

Alteração	Descrição	Alterado em
	<p>Supported Regions (Regiões compatíveis)</p> <ul style="list-style-type: none">• Tipos de nó de cache compatíveis	
Suporte para funções vinculadas ao serviço	<p>Esta versão ElastiCache adicionou suporte para Service Linked Roles (SLR). Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Uso de funções vinculadas ao serviço para o Amazon ElastiCache• Configure suas permissões (somente para novos ElastiCache usuários)	7 de dezembro de 2017
Suporte para tipos de nós R4	<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:</p> <ul style="list-style-type: none">• Tipos de nó de cache compatíveis• Parâmetros específicos de node-type do Redis	20 de novembro de 2017

Alteração	Descrição	Alterado em
ElastiCache para Redis 3.2.10 e suporte para refragmentação online	<p>O Amazon ElastiCache for Redis adiciona suporte ao Redis ElastiCache 3.2.10. ElastiCache for Redis 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:</p> <ul style="list-style-type: none">• Redimensionamento de cluster on-line• Refragmentação e rebalanceamento de fragmentos online para o Redis (modo cluster habilitado)	9 de novembro de 2017
Elegibilidade para HIPAA	<p>ElastiCache para Redis, a versão 3.2.6 agora é certificada para elegibilidade para HIPAA quando a criptografia está habilitada em seu cluster. Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Validação de conformidade para o Amazon ElastiCache• Segurança de dados no Amazon ElastiCache	2 de novembro de 2017

Alteração	Descrição	Alterado em
ElastiCache para Redis 3.2.6 e suporte para criptografia	<p>ElastiCache adiciona suporte ElastiCache para o Redis 3.2.6, que inclui dois recursos de criptografia:</p> <ul style="list-style-type: none">• A criptografia em trânsito criptografa seus dados sempre que eles estão em trânsito, como entre os nós em um cluster ou entre um cluster e o aplicativo.• A criptografia em repouso criptografa dados no disco durante as operações de sincronização e backup. <p>Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Segurança de dados no Amazon ElastiCache• Versões do ElastiCache para Redis com suporte	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 Amazon VPC.</p> <p>Para obter mais informações, consulte Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC no Guia de Usuário ElastiCache</p> <p>.</p>	24 de abril de 2017

Alteração	Descrição	Alterado em
Suporte para testar o failover automático	<p>ElastiCache adiciona suporte para testar o failover automático em clusters Redis que oferecem suporte à replicação. Para obter mais informações, consulte:</p> <ul style="list-style-type: none"> • Teste do failover automático no ElastiCache Guia do usuário. • TestFailover, na Referência de APIs do ElastiCache . • test-failover na AWS CLI Referência. 	4 de abril de 2017
Restauração avançada do Redis	<p>ElastiCache adiciona 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 a API e CLI, esse recurso pode restaurar um número diferente de grupos de nó em vez de um número diferente de fragmentos.) Esta atualização também oferece suporte a diferentes configurações de slot do Redis. Para ter mais informações, consulte Restauração de um backup para um novo cache.</p>	15 de março de 2017
Novo parâmetro Redis de gerenciamento de memória	<p>ElastiCache adiciona um novo parâmetro <code>Redisreserved-memory-percent</code> , que facilita o gerenciamento da memória reservada . Esse parâmetro está disponível em todas as versões do ElastiCache for Redis. Para obter mais informações, consulte:</p> <ul style="list-style-type: none"> • Gerenciamento da memória reservada • Novos parâmetros para o Redis 3.2.4 	15 de março de 2017

Alteração	Descrição	Alterado em
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:</p> <ul style="list-style-type: none">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	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:</p> <ul style="list-style-type: none">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	8 de dezembro de 2016
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:</p> <ul style="list-style-type: none">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	1 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:</p> <ul style="list-style-type: none">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	17 de outubro de 2016

Alteração	Descrição	Alterado em
Suporte para Redis Cluster	<p>ElastiCache adiciona suporte ao Redis Cluster (aprimorado). Os clientes que usam o Redis 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 Cluster são cerca de um quarto dos tempos 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:</p> <ul style="list-style-type: none">• Comparação entre o Memcached e o Redis• ElastiCache para componentes e recursos do Redis: observe as seções sobre Nós, Fragmentos, Clusters e Replicação.• Terminologia do ElastiCache for Redis	12 de outubro de 2016
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:</p> <ul style="list-style-type: none">• Tipos de nó de cache compatíveis• Parâmetros específicos de node-type do Redis	3 de agosto de 2016

Alteração	Descrição	Alterado em
Suporte para a região de Mumbai	<p>ElastiCache adiciona suporte para a região Ásia-Pacífico (Mumbai). Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Tipos de nó de cache compatíveis• Parâmetros específicos de node-type do Redis	27 de junho de 2016
Exportação de snapshots	<p>ElastiCache adiciona a capacidade de exportar um snapshot do Redis para que você possa acessá-lo de fora. ElastiCache Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Exportação de um backup no Guia do ElastiCache usuário da Amazon• CopySnapshot na Amazon ElastiCache API Reference	26 de maio de 2016
Ampliação do tipo de nó	<p>ElastiCache adiciona a capacidade de escalar seu tipo de nó Redis. Para ter mais informações, consulte Dimensionamento ElastiCache para Redis.</p>	24 de março de 2016
Atualização fácil do mecanismo	<p>ElastiCache adiciona a capacidade de atualizar facilmente seu mecanismo de cache Redis. Para ter mais informações, consulte Versões de mecanismos e atualização.</p>	22 de março de 2016
Suporte para tipos de nós R3	<p>ElastiCache adiciona suporte para tipos de nós R3 na região da China (Pequim) e na região da América do Sul (São Paulo). Para obter mais informações, consulte Supported cache node types.</p>	16 de março de 2016

Alteração	Descrição	Alterado em
Acessando ElastiCache usando uma função Lambda	Foi adicionado um tutorial sobre como configurar uma função Lambda para ElastiCache acessar em uma Amazon VPC. Para ter mais informações, consulte Tutoriais e vídeos do ElastiCache .	12 de fevereiro de 2016
Suporte para Redis 2.8.24	ElastiCache adiciona suporte para a versão 2.8.24 do Redis com melhorias adicionadas desde o Redis 2.8.23. Entre os aprimoramentos incluem correções de bugs e suporte para o registro em log de endereços de acesso à memória incorretos. Para obter mais informações, consulte: <ul style="list-style-type: none">• ElastiCache para Redis versão 2.8.24 (aprimorado)• Redis 2.8 Notas de release	20 de janeiro de 2016
Suporte para a região Ásia-Pacífico (Seul).	ElastiCache adiciona suporte para a região Ásia-Pacífico (Seul) (ap-northeast-2) com os tipos de nós t2, m3 e r3.	6 de janeiro de 2016
Mudança ElastiCache no console da Amazon.	Como as versões mais recentes do Redis oferecem uma experiência de usuário melhor e mais estável, 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 Versões do ElastiCache para Redis com suporte .	15 de dezembro de 2015

Alteração	Descrição	Alterado em
Suporte para Redis 2.8.23.	ElastiCache adiciona suporte para a versão 2.8.23 do Redis com melhorias adicionadas desde o Redis 2.8.22. Entre os aprimoramentos estão correções de bugs e suporte para o novo parâmetro <code>close-on-slave-write</code> que, se ativado, desconecta clientes que tentam gravar em uma réplica somente leitura. Para ter mais informações, consulte ElastiCache para Redis versão 2.8.23 (aprimorado) .	13 de novembro de 2015

Alteração	Descrição	Alterado em
Suporte para Redis 2.8.22.	<p>ElastiCache adiciona suporte para a 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">• Implementação de um processo de salvamento sem bifurcação que permite um salvamento bem-sucedido quando a memória disponível baixa pode fazer com que uma gravação com bifurcação falhe.• CloudWatch Métricas adicionais — <code>SaveInProgressReplicationBytes</code>.• Para habilitar sincronizações parciais, o parâmetro Redis <code>repl-backlog-size</code> agora se aplica a todos os clusters. <p>Para obter uma lista completa de mudanças e mais informações, consulte ElastiCache para Redis versão 2.8.22 (aprimorado).</p> <p>Esta versão da documentação inclui uma reorganização da documentação e a remoção da documentação da interface de linha de comando (CLI) de ElastiCache. Para usar a linha de comando, consulte a Linha de AWS comando para ElastiCache.</p>	28 de setembro de 2015
Suporte para Redis 2.8.21	ElastiCache adiciona suporte para a versão 2.8.21 do Redis e melhorias do Redis desde a versão 2.8.19. Esta versão do Redis inclui diversas correções de bugs. Para obter mais informações, consulte as Notas de release do Redis 2.8 .	29 de julho de 2015

Alteração	Descrição	Alterado em
Novo tópico: Acessando ElastiCache de fora AWS	Foi adicionado um novo tópico sobre como acessar ElastiCache recursos de fora AWS. Para obter mais informações, consulte Acessando ElastiCache de fora AWS .	9 de julho de 2015
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 Notificações de eventos e o Amazon SNS.</p>	11 de junho de 2015

Alteração	Descrição	Alterado em
Suporte para Redis v.2.8.19.	<p>ElastiCache adiciona suporte para a versão 2.8.19 do Redis e melhorias do Redis desde a versão 2.8.6. Este suporte inclui suporte para:</p> <ul style="list-style-type: none">• A estrutura HyperLogLog de dados, com os comandos Redis PFADD, PFCOUNT e PFMERGE.• Consultas de alcance lexicográfico com os novos comandos ZRANGEBYLEX, ZLEXCOUNT e ZREMRANGEBYLEX.• Uma série de correções de erros, impedindo que um nó primário envie dados obsoletos aos nós de réplica, falhando o SYNC primário quando um processo filho de salvamento em segundo plano (bgsave) é encerrado inesperadamente. <p>Para obter mais informações sobre HyperLogLog, consulte a nova estrutura de dados do Redis: a. HyperLogLog</p> <p>Para obter mais informações sobre PFADD, PFCOUNT e PFMERGE, consulte a documentação do Redis e clique em. HyperLogLog</p>	11 de março de 2015
Suporte para tags de alocação de custos	<p>ElastiCache adiciona suporte para tags de alocação de custos. Para ter mais informações, consulte Monitoramento de custos com tags de alocação de custos.</p>	9 de fevereiro de 2015
Support para a região AWS GovCloud (Oeste dos EUA)	<p>ElastiCache adiciona suporte para a região AWS GovCloud (Oeste dos EUA) (us-gov-west-1).</p>	29 de janeiro de 2015

Alteração	Descrição	Alterado em
Suporte para a região Europa (Frankfurt)	ElastiCache adiciona suporte para a região Europa (Frankfurt) (eu-central-1).	19 de janeiro de 2015
Suporte a Multi-AZ para grupos de replicação do Redis	ElastiCache adiciona suporte para Multi-AZ do nó primário a uma réplica de leitura em um grupo de replicação do Redis. 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 ter mais informações, consulte Minimização do tempo de inatividade no ElastiCache para Redis com o multi-AZ .	24 de outubro de 2014
AWS CloudTrail I registro de chamadas de API suportado	ElastiCache adiciona suporte para uso AWS CloudTrail para registrar todas as chamadas de ElastiCache API. Para ter mais informações, consulte Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail .	15 de setembro de 2014
Novos tamanhos de instâncias com suporte	ElastiCache adiciona suporte para instâncias adicionais de uso geral (T2). Para ter mais informações, consulte Configuração de parâmetros do mecanismo usando grupos de parâmetros .	11 de setembro de 2014
Novos tamanhos de instâncias com suporte	ElastiCache adiciona suporte para instâncias adicionais de uso geral (M3) e instâncias otimizadas para memória (R3). Para ter mais informações, consulte Configuração de parâmetros do mecanismo usando grupos de parâmetros .	1 de julho de 2014

Alteração	Descrição	Alterado em
Backup e restauração para clusters do Redis	Nesta versão, ElastiCache permite que os clientes criem instantâneos de seus 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 todos os dados no cache do Redis. 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 ter mais informações, consulte Snapshots e restauração .	24 de abril de 2014
Redis 2.8.6	ElastiCache suporta o Redis 2.8.6, além do Redis 2.6.13. Com o Redis 2.8.6, os clientes podem melhorar a resiliência e a tolerância a falhas de réplicas de leitura, com suporte para ressincronização parcial e um número mínimo definido pelo usuário de réplicas de leitura que devem estar disponíveis em todos os momentos. O Redis 2.8.6 também oferece suporte completo para publish-and-subscribe, onde os clientes podem ser notificados sobre eventos que ocorrem no servidor.	13 de março de 2014

Alteração	Descrição	Alterado em
Mecanismo de cache Redis	<p>ElastiCache oferece o software de mecanismo de cache Redis, além do Memcached. Os clientes que atualmente usam o Redis podem “semear” um novo cluster de cache do ElastiCache Redis com seus dados existentes a partir de um arquivo de instantâneo do Redis, facilitando a migração para um ambiente gerenciado. ElastiCache</p> <p>Para oferecer suporte aos recursos de replicação do Redis, a ElastiCache API agora oferece suporte a grupos de replicação. Os clientes podem criar um grupo de replicação com um nó de cache Redis primário e adicionar um ou mais nós de réplica de leitura que permanecem automaticamente sincronizados com dados de cache no nó primário. Aplicativos que exigem leitura intensa podem ser descarregados para uma réplica de leitura, reduzindo a carga no nó primário. Réplicas de leitura também podem proteger contra a perda de dados no caso de uma falha do nó de cache primário.</p>	3 de setembro de 2013
Suporte para a Amazon Virtual Private Cloud (VPC) padrão.	<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 ter mais informações, consulte Segurança do Amazon VPCs e do ElastiCache.</p>	8 de janeiro de 2013

Alteração	Descrição	Alterado em
Suporte para a Amazon Virtual Private Cloud (VPC).	Nesta versão, os ElastiCache clusters podem ser lançados na Amazon Virtual Private Cloud (VPC). Por padrão, clusters de cache de novos clientes são criados em uma Amazon VPC automaticamente, enquanto os clientes existentes podem migrar para a Amazon VPC ao seu próprio ritmo. Para ter mais informações, consulte Segurança do Amazon VPCs e do ElastiCache .	20 de dezembro 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.