



ElastiCache Guia do usuário do Memcached

Amazon ElastiCache



Versão da API 2015-02-02

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

Amazon ElastiCache: ElastiCache Guia do usuário do Memcached

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 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 Memcached?	1
Armazenamento em cache sem servidor	1
Clusters autoprotetidos	2
Como funciona	2
Cache e mecanismos de cache	2
Casos de uso	8
Armazenamento de dados na memória	8
Depoimentos de clientes do ElastiCache	10
Recursos do ElastiCache for Memcached	10
Ferramentas para o gerenciamento da sua implementação	12
Usar a AWS Management Console	12
Usar a AWS CLI	12
Uso da SDK AWS	12
Uso da API do ElastiCache	12
Consulte também	13
Escolha entre opções de implantação	13
Comparação dos caches autoprotetidos do Memcached e do Redis	14
Conceitos básicos do ElastiCache for Memcached	20
Configuração	20
Inscreva-se para um Conta da AWS	20
Criar um usuário administrativo	21
Conceder acesso programático	22
Configurar permissões	24
Configurar o EC2	25
Conceder acesso de rede	25
Criar um cache	26
Criar um cache sem servidor	26
Ler e gravar dados	27
Limpar (opcional)	32
Próximos Passos	33
Tutorial: configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC	34
Etapa 1: criar um cache do ElastiCache	34
Etapa 2: criar uma função do Lambda	36

Etapa 3: testar a função do Lambda	40
Tutoriais e vídeos	42
Vídeos	43
Escolher regiões e zonas de disponibilidade	49
Considerações sobre Zonas de disponibilidade	50
Regiões e endpoints com suporte	52
Localização dos seus nós	58
Uso de zonas locais	58
Ativação de uma zona local	59
Uso do Outposts	59
Uso de Outposts com o console do Memcached	60
Uso de Outposts com a CLI da AWS	62
Projeto do próprio cluster do ElastiCache	63
ElastiCache para componentes e recursos do Memcached	64
Nodes	64
Clusters	65
AWS Regiões e zonas de disponibilidade	66
Endpoints	67
Grupos de parâmetros	67
Segurança	68
Grupos de sub-redes	68
Eventos	68
Gerenciamento de clusters	69
Escolher um tipo de rede	71
Descoberta automática	73
Preparação de um cluster	116
Criar um cluster	123
Visualização dos detalhes de um cluster	126
Modificar um cluster	131
Reinicialização de um cluster	135
Adição de nós a um cluster	137
Remoção de nós de um cluster	143
Cancelamento de operações de adição ou exclusão de nós pendentes	149
Excluir um cluster	150
Acesso ao seu cluster	153
Localização de endpoints de conexão	159

Gerenciamento de nós	167
Visualizando o status ElastiCache do nó	167
Conexão a nós	173
Tipos de nó compatíveis	176
Substituição de nós	186
Nós reservados	188
Migração de nós da geração anterior	200
Trabalhando com ElastiCache	202
Versões de mecanismos e atualização	202
Versões do Memcached com suporte	203
Versões de mecanismos e atualização	207
Como atualizar as versões dos mecanismos	208
Práticas recomendadas e estratégias de armazenamento em cache	210
Práticas recomendadas com clientes do Memcached	210
Comandos do Memcached compatíveis	218
Estratégias de armazenamento em cache	219
Gerenciamento do cluster autoprojetoado	224
Gerenciamento da manutenção	225
Configuração de parâmetros do mecanismo usando grupos de parâmetros	227
Dimensionamento ElastiCache para Memcached	269
Dimensionamento ElastiCache para Memcached	269
Definição dos limites de escalabilidade para gerenciar custos	269
Pré-escalamento com Serverless ElastiCache	269
Definindo limites de escala usando o console e AWS CLI	271
Dimensionamento ElastiCache para clusters autoprojetoados pelo Memcached	272
Marcação de recursos do ElastiCache	276
Monitoramento de custos com tags	283
Gerenciar tags usando o AWS CLI	285
Gerenciamento de tags usando a API do ElastiCache	288
Lente do Well-Architected para o Amazon ElastiCache	291
Pilar Excelência operacional	291
Pilar Segurança	300
Pilar Confiabilidade	307
Pilar Eficiência de performance	313
Pilar Otimização de custos	323
Solução de problemas	330

Problemas de conectividade	330
Erros do cliente Redis	331
Solução de problemas de alta latência no Serverless ElastiCache	332
Solução de problemas de limitação no Serverless ElastiCache	333
Related Topics	334
Etapas adicionais de solução de	334
Grupos de segurança	335
Network ACLs	335
Tabelas de rotas	337
Resolução do DNS	337
Identificação de problemas com o diagnóstico do lado do servidor	338
Validação da conectividade de rede	344
Limites relacionados à rede	346
Uso da CPU	347
Conexões sendo encerradas do lado do servidor	350
Solução de problemas no lado do cliente para instâncias do Amazon EC2	352
Dissecação do tempo necessário para concluir uma única solicitação	353
Segurança	356
Proteção de dados	357
Segurança de dados no Amazon ElastiCache	357
Privacidade do tráfego entre redes	367
Segurança do Amazon VPCs e do ElastiCache	367
API do Amazon ElastiCache e endpoints da VPC de interface (AWS PrivateLink)	392
Sub-redes e grupos de sub-redes	395
Identity and Access Management	403
Público	403
Autenticando com identidades	404
Como gerenciar acesso usando políticas	408
Como o Amazon ElastiCache funciona com o IAM	410
Exemplos de políticas baseadas em identidade	418
Solução de problemas	421
Controle de acesso	423
Visão geral do gerenciamento de acesso	424
Validação de compatibilidade	455
Mais informações	457
Resiliência	457

Atenuar falhas	457
Segurança da infraestrutura	460
Atualizações de serviço	460
Gerenciando atualizações de serviços	460
Registrar em log e monitoramento	467
Métricas e eventos sem servidor	467
Métricas sem servidor	467
Eventos sem servidor	471
Métricas e eventos autoprojeados	476
Métricas de cluster autoprojetoado	476
Eventos de cluster autoprojeados	476
Monitoramento do uso	484
Monitoramento de eventos do Amazon SNS	499
Logs das chamadas de API do Amazon ElastiCache com AWS CloudTrail	516
Informações sobre o Amazon ElastiCache no CloudTrail	516
Noções básicas sobre entradas de arquivo de log do Amazon ElastiCache	517
Cotas	522
Referência	523
Uso da API do ElastiCache	523
Como usar a API de consulta	523
Bibliotecas disponíveis	527
Solução de problemas de aplicações	527
Configurar a AWS CLI para o ElastiCache	528
Pré-requisitos	529
Obtenção das ferramentas da linha de comando	531
Configuração das ferramentas	531
Como fornecer credenciais para as ferramentas	532
Variáveis de ambiente	533
Mensagens de erro	534
Notificações	536
Notificações gerais do ElastiCache	536
Notificações do ElastiCache for Memcached	536
.....	537
AWS Glossário	553
.....	dliv

O que é o Amazon ElastiCache for Memcached?

Bem-vindo ao Guia do usuário do Amazon ElastiCache for Memcached. 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 autoprojetados do Memcached e do Redis](#) neste guia.

Armazenamento em cache sem servidor

ElastiCache for Memcached oferece armazenamento em cache sem servidor, o que simplifica a adição e a operação de um cache baseado em Memcached para seu aplicativo. ElastiCache para Memcached 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 A tecnologia sem servidor também elimina a necessidade de planejar e gerenciar a capacidade de armazenamento em cache. ElastiCache monitora constantemente a memória e a computação do cache usadas pelo seu aplicativo e dimensiona automaticamente a capacidade para atender às necessidades do seu aplicativo. ElastiCache oferece uma experiência de endpoint simples para desenvolvedores, abstraindo a infraestrutura de cache e o software subjacentes. ElastiCache gerencia o provisionamento de hardware, o monitoramento, a substituição de nós e a aplicação de patches de software de forma automática e transparente, para que você possa se concentrar no desenvolvimento de aplicativos, em vez de operar o cache.

ElastiCache for Memcached Serverless é compatível com Memcached 1.6 e superior.

Projetando seu próprio cluster ElastiCache para o Memcached

Se você precisar de um controle refinado sobre seu cluster ElastiCache for Memcached, você pode optar por criar seu próprio cluster Memcached 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 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 o Memcached oferece maior flexibilidade e controle sobre seus clusters. Por exemplo, você pode optar por operar um cluster com disponibilidade single-AZ ou disponibilidade cross-AZ, dependendo das necessidades. 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 Memcached.

Ao criar o seu próprio ElastiCache para o Memcached, você pode optar por executar o Memcached 1.4 ou superior.

Como funciona

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

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 Memcached 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 Memcached](#)
- [Dimensões de preço](#)

Como funciona ElastiCache o Memcached

ElastiCache para Memcached Serverless

ElastiCache para Memcached 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 Memcached para começar a acessar o cache.

Note

ElastiCache for Memcached Serverless só é compatível com clientes Memcached que oferecem suporte a TLS.

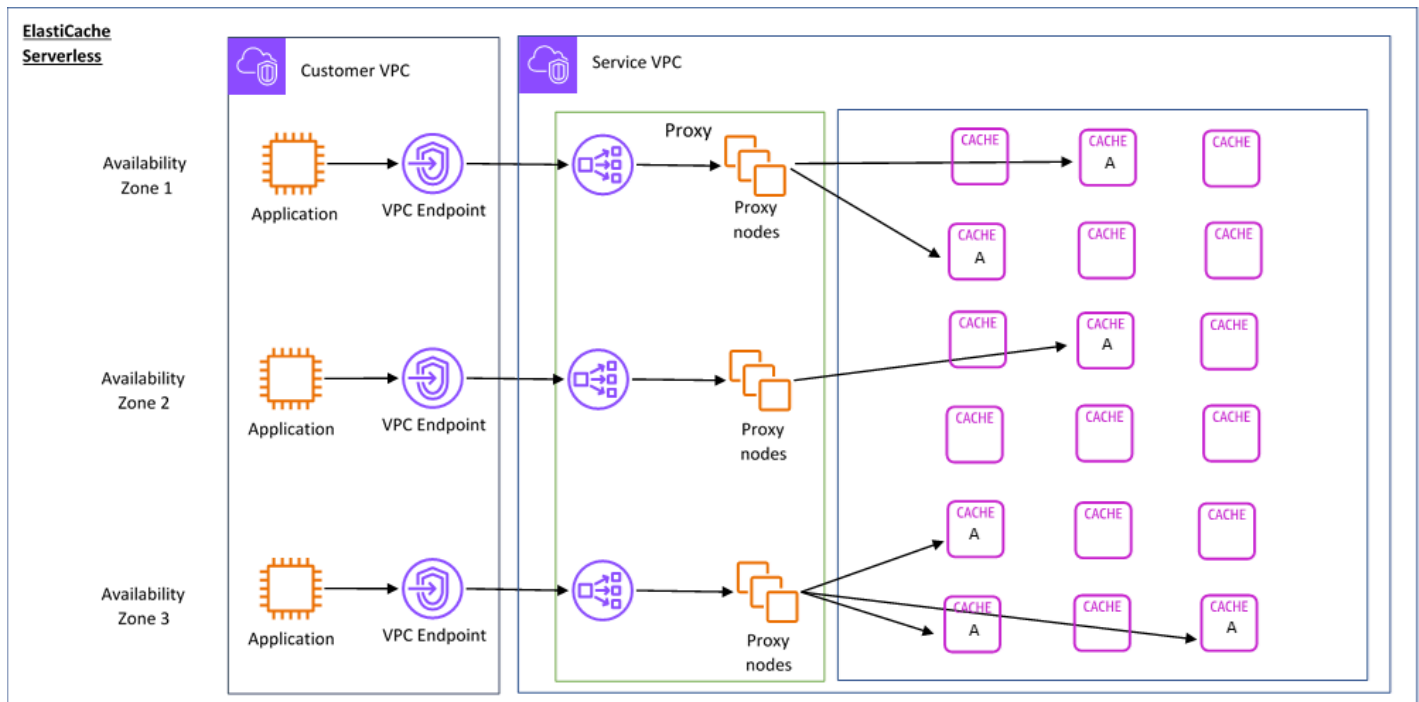
Benefícios principais

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

disponibilidade do seu aplicativo. Quando uma nova versão principal do Memcached 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.

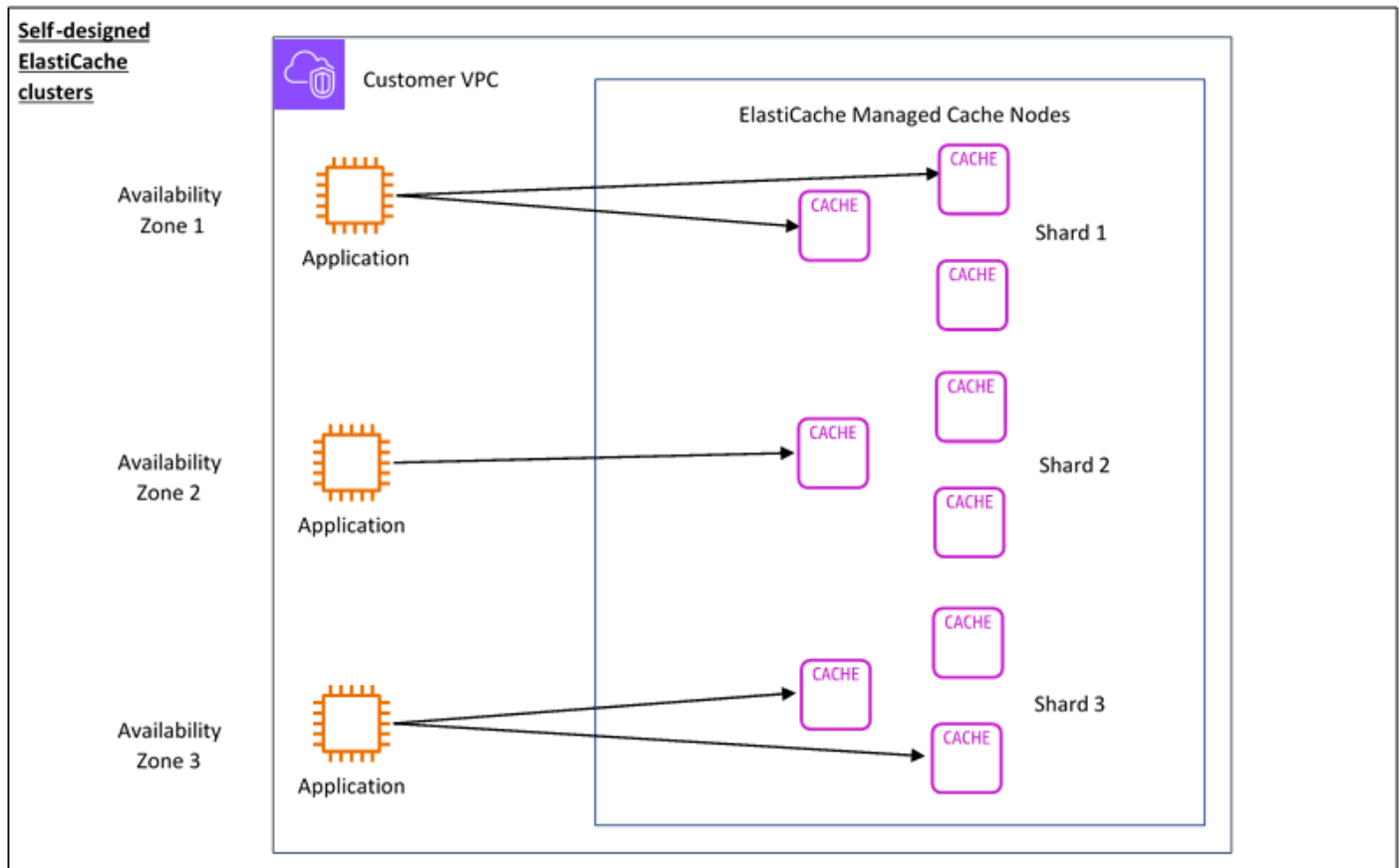
Clusters autoprojeto ElastiCache

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 oferece um controle mais refinado sobre a configuração e a escala do cluster.

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 [Configuração de parâmetros do mecanismo usando grupos de parâmetros](#) 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 escalar horizontalmente adicionando nós.

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



Dimensões de preço

Você pode implantar ElastiCache em duas opções de implantação. Ao implantar o ElastiCache Serverless, você paga pelo uso dos dados armazenados em GB/hora e pela computação em ElastiCache unidades de processamento (ECPUs). Ao escolher criar seus próprios clusters ElastiCache para o Memcached, você paga por hora de uso do nó de cache. Consulte detalhes de preço [aqui](#).

Armazenamento de dados

Você paga pelos dados armazenados no ElastiCache Serverless cobrados em gigabytes por hora (GB-hrs). ElastiCache O Serverless monitora continuamente os dados armazenados em seu cache, fazendo amostragens várias vezes por minuto, e calcula uma média horária para determinar o uso do armazenamento de dados do cache em GB-h. Cada cache ElastiCache sem servidor é medido para um mínimo de 1 GB de dados armazenados.

ElastiCache Unidades de processamento (ECPUs)

Você paga pelas solicitações que seu aplicativo executa 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 quilobyte (KB) de dados transferidos. Por exemplo, um comando GET que transfere até 1 KB de dados consome uma ECPU. Uma solicitação SET que transfere 3,2 KB de dados vai consumir 3,2 eCPUs.
- Comandos que atuam em vários itens vão consumir proporcionalmente mais eCPUs. Por exemplo, se realizar um multiget em três itens, a aplicação vai consumir três eCPUs.
- Comandos que atuam em mais itens e transferem mais dados consomem eCPUs com base na maior das duas dimensões. Por exemplo, se a aplicação usar o comando GET, recuperar três itens e transferir 3,2 KB de dados, ela consumirá 3,2 ECPU. Como alternativa, se 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 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.

Casos de uso comuns do ElastiCache e como ele pode ajudar

Seja para fornecer as últimas notícias ou 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.

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 Memcached

Depoimentos de clientes do ElastiCache

Para saber mais sobre como empresas como a Airbnb, a PBS e 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.

Recursos do ElastiCache for Memcached

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

- Destaques do serviço e definição de preços: a [página de detalhes do produto](#) fornece uma visão geral do ElastiCache, destaques de serviços e preços.
- Vídeos: a seção [Vídeos do ElastiCache](#) tem vídeos que o apresentam ao Amazon ElastiCache for Memcached, abordam casos de uso comuns e demonstram como usar o ElastiCache for Memcached para reduzir a latência e melhorar o throughput de suas aplicações.
- Conceitos básicos: a seção [Conceitos básicos do Amazon ElastiCache for Memcached](#) inclui um exemplo que o demonstra passo a passo o processo de criar um cluster de cache, 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 permitem que sua aplicação apresente 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) compatíveis com 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.

Ferramentas para o gerenciamento da sua implementação

Quando tiver concedido à sua instância do Amazon EC2 acesso ao seu cluster do ElastiCache, você terá quatro meios com os quais pode gerenciar seu cluster do ElastiCache: o AWS Management Console , a AWS CLI para o ElastiCache, o SDK da AWS para ElastiCache e a API do ElastiCache.

Usar a AWS Management Console

O AWS Management Console é a maneira mais fácil de gerenciar o Amazon ElastiCache for Memcached. O console permite criar clusters de cache, adicionar e remover nós de cache e realizar outras tarefas administrativas sem ter que escrever nenhum código. O console também fornece gráficos de performance do nó de cache do CloudWatch, mostrando a atividade do mecanismo de cache, a utilização de memória e CPU, bem como outras métricas. Para obter mais informações, consulte tópicos específicos neste Guia do usuário.

Usar a AWS CLI

Você também pode usar a AWS Command Line Interface (AWS CLI) para o ElastiCache. A AWS CLI facilita a realização de uma operação de cada vez, como iniciar ou interromper um cluster de cache. Você também pode invocar comandos da AWS CLI para o ElastiCache a partir de uma linguagem de scripts de sua escolha, permitindo automatizar tarefas repetidas. Para obter mais informações sobre a AWS CLI, consulte o Guia do usuário e a [Referência de comandos da AWS CLI](#).

Uso da SDK AWS

Se quiser acessar o ElastiCache a partir de uma aplicação, você pode usar um dos kits de desenvolvimento de software (SDKs) da AWS. Os SDKs envolvem o chamadas da API do ElastiCache e isolam sua aplicação dos detalhes de baixo nível da API do ElastiCache. Você fornece suas credenciais, e essas bibliotecas de SDK cuidarão da assinatura das solicitações e das autenticações. Para obter mais informações sobre como usar os AWS SDKs, consulte [Ferramentas para a Amazon Web Services](#).

Uso da API do ElastiCache

Você também pode escrever o código da aplicação diretamente com base na API de serviço da Web do ElastiCache. Ao usar a API, você deve escrever o código necessário para construir e autenticar suas solicitações HTTP, analisar os resultados do ElastiCache e lidar com todos os erros. Para obter mais informações sobre a API, consulte [Uso da API do ElastiCache](#).

Consulte também

Para obter mais informações detalhadas sobre o gerenciamento da implantação do Amazon ElastiCache for Memcached, consulte o seguinte:

- [Trabalhando com ElastiCache](#)
- [Privacidade do tráfego entre redes](#)
- [Registros e monitoramento no Amazon ElastiCache](#)

Escolha entre opções de implantação

O Amazon ElastiCache tem duas opções de implantação:

- Armazenamento em cache sem servidor
- Projetar o próprio cluster

Armazenamento em cache sem servidor

O Amazon ElastiCache Sem Servidor simplifica a criação de cache e escala instantaneamente para dar suporte às aplicações mais exigentes dos clientes. Com o ElastiCache Sem Servidor, 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. O ElastiCache Sem Servidor armazena automaticamente os dados de maneira redundante em três zonas de disponibilidade e oferece um [Acordo de Serviço \(SLA\)](#) com 99,99% de disponibilidade. O ElastiCache oferece replicação automática de dados em AZs, eliminando a necessidade de gerenciar manualmente réplicas e software personalizado para mantê-las sincronizadas.

Clusters autoprojetados

Se precisar de um controle refinado sobre o cluster do ElastiCache para Memcached, você poderá optar por criar o próprio cluster do Memcached com o ElastiCache. 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.

Escolha entre opções de implantação

Escolha o armazenamento sem servidor se:

- Você está criando um novo cache para workloads novas ou desconhecidas
- Você tem tráfego de aplicação imprevisível
- Você deseja a maneira mais fácil de começar a usar um cache

Opte por projetar o próprio cluster do ElastiCache se:

- Você já estiver executando o ElastiCache Sem Servidor e deseja um controle mais refinado sobre o tipo de nó que executa o Memcached, o número de nós e o posicionamento dos nós.
- Você não espera que o tráfego da aplicação flutue muito ou pode prever com precisão os altos e baixos do tráfego da aplicação.
- Você pode prever os requisitos de capacidade para controlar os custos.

Tópicos relacionados:

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

Comparação dos caches autoprojeto 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 várias 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](#).

Conceitos básicos do Amazon ElastiCache for Memcached

Os tópicos nesta seção o orientam no processo de criação, de concessão de acesso, de conexão e por fim de exclusão de um cache sem servidor do Memcached usando o console do ElastiCache.

Tópicos

- [Configuração](#)
- [Etapa 1: criar um cache](#)
- [Etapa 2: ler e gravar dados no cache](#)
- [Etapa 3: \(opcional\) limpar](#)
- [Próximos Passos](#)
- [Tutorial: configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC](#)
- [Tutoriais e vídeos do ElastiCache](#)

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](#)

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 .

2. 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. • Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de

Qual usuário precisa de acesso programático?	Para	Por
		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	Siga as instruções da interface que deseja utilizar. <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 para Memcached usa as portas 11211 e 11212 para aceitar comandos do Memcached. Para conectar e executar com êxito os comandos do Memcached a partir da instância do EC2, o grupo de segurança deve permitir acesso a essas portas.

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 11211.
 - 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 11212.
- 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

Etapa 1: criar um cache

O cache que você está prestes a iniciar estará ativo, e não em execução em um sandbox. Você incorrerá nas taxas de uso do ElastiCache padrão pelo cache até ele ser excluído. As cobranças totais serão mínimas (normalmente inferiores a um dólar) se você concluir o exercício descrito aqui em uma única sessão e excluir o cache quando terminar. Para obter mais informações sobre taxas de uso do ElastiCache, consulte [Amazon ElastiCache](#).

Criar um cache sem servidor

AWS Management Console

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

Para criar um novo cache usando a AWS CLI

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

Linux

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

Windows

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

O valor do campo Status está definido como CREATING.

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

Linux

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

Windows

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

Depois de criar o novo cache, avance até [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](#).

Por padrão, ElastiCache cria um cache na sua VPC padrão. Verifique se a instância do EC2 também foi criada na VPC padrão, para que ela possa se conectar ao cache.

Configuração

Antes de começar, verifique se você tem as portas certas disponíveis para acesso.

Porta primária: 11211

Porta otimizada para leitura: 11212

Os caches do Memcached sem servidor anunciam duas portas com o mesmo nome de host. A porta primária permite gravações e leituras com as mesmas garantias de consistência do OSS Memcached. A porta otimizada para leitura permite gravações e, além disso, leituras eventualmente consistentes de menor latência.

Encontre o endpoint do cache

AWS Management Console

Para encontrar o endpoint do seu cache usando o ElastiCache 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, no lado esquerdo do console, escolha Caches do Memcached.
3. No lado direito do console, clique no nome do cache que você acabou de criar.
4. Nos Detalhes do cache, localize e copie o endpoint do cache.

AWS CLI

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

Linux

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

Windows

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

Conectar usando OpenSSL

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

Conectar usando o cliente Java do Memcached

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

Conectar usando o cliente PHP do Memcached

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

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

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

Conectar usando o cliente Python do Memcached (Pymemcache)

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

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

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

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

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

Instalar por meio de npm `i memcache-client`

Na aplicação, crie um cliente TLS do Memcached assim:

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

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

Consulte <https://crates.io/crates/memcache> e <https://github.com/aisk/rust-memcache>.

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

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

Conectar usando o cliente Go do Memcached (Gomemcache)

Consulte <https://github.com/bradfitz/gomemcache>

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

Conectar usando o cliente Ruby do Memcached (Dalli)

Consulte <https://github.com/petergoldstein/dalli>

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

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

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

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

Agora você já pode avançar para [Etapa 3: \(opcional\) limpar](#).

Etapa 3: (opcional) limpar

Usar o AWS Management Console

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

Para excluir um cache

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

Important

Você só pode excluir um cache de cada vez no console do ElastiCache. Escolher vários caches desabilita a operação de exclusão.

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

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

Usar o AWS CLI

O código a seguir exclui o cache my-cache.

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

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

Para Linux, macOS ou Unix:

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

Para Windows:

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

Para obter mais informações, consulte a AWS CLI em busca do tópico do ElastiCache `delete-serverless-cache`.

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

Próximos Passos

Para obter mais informações sobre o ElastiCache, consulte:

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

Tutorial: configurar uma função do Lambda para acessar o Amazon ElastiCache em uma Amazon VPC

Neste tutorial, você faz o seguinte:

- Crie um cache do Amazon ElastiCache no Amazon Virtual Private Cloud (Amazon VPC) padrão na região us-east-1.
- Crie uma função do Lambda para acessar o cache do ElastiCache. Ao criar a função do Lambda, você fornece os IDs de sub-rede da sua Amazon VPC e grupo de segurança de VPC para permitir que a função do Lambda acesse os recursos na sua VPC. Para ilustração neste tutorial, a função do Lambda gera um UUID, o grava em cache e o recupera do cache.
- Invoque a função do Lambda manualmente e verifique se ela acessou o cache do ElastiCache na sua VPC.

Important

O tutorial usa a Amazon VPC padrão na região us-east-1 da sua conta. Para obter mais informações sobre uma Amazon VPC, consulte [How to Get Started with Amazon VPC](#) (Conceitos básicos da Amazon VPC) e o Guia do usuário da Amazon VPC.

Tópicos

- [Etapa 1: criar um cache do ElastiCache](#)
- [Etapa 2: criar uma função do Lambda](#)
- [Etapa 3: testar a função do Lambda](#)

Conceitos básicos

[Etapa 1: criar um cache do ElastiCache](#)

Etapa 1: criar um cache do ElastiCache

Nesta etapa, crie um cache do Amazon ElastiCache no Amazon Virtual Private Cloud padrão na região us-east-1 em sua conta usando a AWS CLI. Para obter informações sobre a criação de caches sem servidor do ElastiCache usando o console ou a API do ElastiCache, consulte [Criar um cluster](#) no Guia do usuário do ElastiCache para Memcached.

AWS Management Console

Execute o comando da AWS CLI a seguir para criar um novo cluster sem servidor do Memcached na VPC padrão na região us-east-1 da sua conta.

Linux

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

Windows

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

O valor do campo Status está definido como CREATING. Pode levar alguns minutos para que o ElastiCache conclua a criação do cluster.

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 serverlessCacheforLambda \  
--region us-east-1
```

Windows

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

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

Depois de criar o novo cache, avance até [Etapa 2: criar uma função do Lambda](#).

Próxima etapa:

[Etapa 2: criar uma função do Lambda](#)

Etapa 2: criar uma função do Lambda

Nesta etapa, faça o seguinte:

1. Crie um pacote de implantação de uma função do Lambda usando o código de exemplo fornecido.
2. Crie uma função do IAM (função de execução). Quando fizer upload do pacote de implantação, precisará especificar essa função, de forma que o Lambda consiga assumir a função e executá-la em seu nome. A política de permissões concede permissões ao AWS Lambda para configurar interfaces de rede elástica (ENIs) para permitir que sua função do Lambda acesse recursos na VPC. Nesse exemplo, sua função do Lambda acessa um cluster do ElastiCache na VPC.
3. Crie a função do Lambda; fazendo upload do pacote de implantação.

Próxima etapa

[Etapa 2.1: criar o pacote de implantação](#)

Etapa 2.1: criar o pacote de implantação

Atualmente, o código de exemplo para a função do Lambda só é fornecido em Python.

Python

O exemplo a seguir do código Python lê e grava um item no seu cluster do ElastiCache. Copie o código e o salve em um arquivo chamado `app.py`. Não se esqueça de substituir o valor `elasticache_config_endpoint` no código pelo endereço do endpoint que você copiou na etapa 1.

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

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

context = ssl.create_default_context()
```

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

def lambda_handler(event, context):

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

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

    # get the item (UUID) from the cache
    result = memcached_client.get("uuid")
    decoded_result = result.decode("utf-8")

    # check the retrieved item matches the item added to the cache and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Memcached.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
        {decoded_result}")

    return "Fetched value from Memcached"
```

Esse código usa a biblioteca [pymemcache](#) Python para colocar itens no cache e recuperá-los. Para criar um pacote de implantação contendo pymemcache, realize as etapas a seguir.

1. No diretório do projeto que contém o arquivo de código-fonte `app.py`, crie uma pasta `package` na qual instalar a biblioteca `pymemcache`.

```
mkdir package
```

2. Instale `pymemcache` usando `pip`.

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

3. Crie um arquivo `.zip` contendo a biblioteca `pymemcache`. No Linux e no MacOS, execute o comando a seguir. No Windows, use o utilitário `zip` preferencial para criar um arquivo `.zip` com a biblioteca `pymemcache` na raiz.

```
cd package
```

```
zip -r ../my_deployment_package.zip .
```

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

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

Próxima etapa

[Etapa 2.2: Criar o perfil do IAM \(perfil de execução\)](#)

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

Nesta etapa, você cria um perfil do AWS Identity and Access Management (IAM) usando o tipo de perfil e a política de acesso predefinidos abaixo:

- Perfil de serviço da AWS do tipo AWS Lambda: esse perfil concede permissões à AWS para assumir o perfil de serviço.
- **AWSLambdaVPCLambdaAccessExecutionRole**: esta é a política de permissões de acesso que você anexa ao perfil. A política concede permissões para as ações do EC2 que o AWS Lambda precisa para gerenciar ENIs. Você pode visualizar essa política gerenciada pela AWS no console do IAM.

Para obter mais informações sobre perfis do IAM, consulte [Funções \(delegação e federação\)](#) no Guia do usuário do IAM.

Use o procedimento a seguir para criar a função do IAM.

Como criar um perfil do IAM (de execução).

1. Faça login no Console de Gerenciamento da AWS e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Escolha Roles (Perfis) e, depois, Create Role (Criar perfil).
 - Em Tipo de entidade de confiança, escolha Serviço AWS e, em Casos de uso, escolha Lambda. Isso concede ao AWS Lambda permissões de serviço para assumir a função. Escolha Next (Próximo).
 - Em Adicionar permissões, procure **AWSLambdaVPCLambdaAccessExecutionRole** e marque a caixa de seleção próxima da política.

- Escolha Next (Próximo).
 - Em Role Name (Nome de função), use um nome que seja exclusivo dentro da sua conta da AWS (por exemplo, lambda-vpc-execution-role).
 - Selecione Criar perfil.
3. Copie o ARN da função. Você precisará dele na próxima etapa ao criar sua função do Lambda.

Próxima etapa

[Etapa 2.3: fazer upload do pacote de implantação \(criar função do Lambda\)](#)

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

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

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

Para a opção `role`, use o ARN da função de execução criada na etapa 2.2. Para `vpc-config`, insira listas separadas por vírgulas das sub-redes da VPC padrão e o ID do grupo de segurança da VPC padrão. É possível encontrar esses valores no [console do Amazon VPC](#). Para encontrar as sub-redes da VPC padrão, escolha As VPCs e a VPC padrão da Conta da AWS. Para encontrar o grupo de segurança dessa VPC, em Segurança, escolha Grupos de segurança. Não se esqueça de selecionar a região `us-east-1`.

Para Linux, macOS ou Unix:

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

Para Windows:

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

Opcionalmente, você pode fazer upload do arquivo .zip a um bucket do Amazon S3 na mesma região da AWS e, em seguida, especificar o bucket e o nome do objeto no comando anterior. Você deve substituir o parâmetro `--zip-file` pelo parâmetro `--code`, conforme mostrado a seguir.

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

Você também pode criar a função do Lambda usando o console do AWS Lambda. Ao criar a função, escolha um VPC para o Lambda e, em seguida, selecione as sub-redes e os grupos de segurança nos campos fornecidos.

Próxima etapa

[Etapa 3: testar a função do Lambda](#)

Etapa 3: testar a função do Lambda

Nesta etapa, você invoca a função do Lambda manualmente usando o comando `invoke`. Quando a função do Lambda for executada, ela gerará um UUID e o gravará no cluster do ElastiCache que você especificou no seu código do Lambda. Depois, a função do Lambda recupera o item do cache.

1. Invoque a função do Lambda (AccessMemCache) usando o comando `invoke` do AWS Lambda.

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws lambda invoke ^  
--function-name AccessMemCache ^  
--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 no CloudWatch Logs abrindo o console do [CloudWatch](#) e escolhendo o grupo de logs da função (/aws/lambda/AccessMemcached). O fluxo de logs deve conter uma saída semelhante à mostrada a seguir:

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

- Analisar os resultados no console do AWS Lambda.

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 \(re:Invent 2016\)](#)
- [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 \(re:Invent 2016\)](#)

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 \(AWS re:Invent 2013\)](#).

Escolher regiões e zonas de disponibilidade

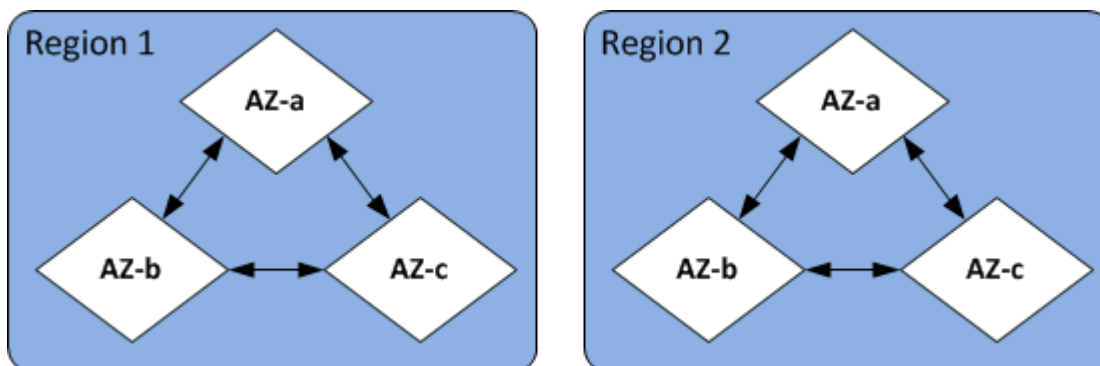
AWS Os recursos de computação em nuvem estão alojados em instalações de data center altamente disponíveis. Para fornecer escalabilidade e confiabilidade adicionais, estas instalações do datacenter estão localizadas em diferentes locais físicos. Esses locais são categorizados por regiões e zonas de disponibilidade.

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

- [Considerações sobre Zonas de disponibilidade](#)
- [Regiões e endpoints com suporte](#)
- [Localização dos seus nós](#)
- [Uso de zonas locais com o ElastiCache](#)

- [Uso do Outposts](#)

Considerações sobre Zonas de disponibilidade

A distribuição de nós do Memcached em várias zonas de disponibilidade em uma região ajuda a protegê-los contra o impacto de uma falha catastrófica, como perda de energia em uma zona de disponibilidade.

Armazenamento em cache sem servidor

ElastiCache o armazenamento em cache sem servidor cria um cache altamente disponível que abrange várias zonas de disponibilidade. Você pode especificar sub-redes de diferentes zonas de disponibilidade e da mesma VPC ao criar seu cache ou ElastiCache escolher sub-redes automaticamente da sua VPC padrão.

Projetando seu próprio cluster ElastiCache para o Memcached

Um cluster do Memcached pode ter até 300 nós. Ao criar ou adicionar nós ao seu cluster Memcached, você pode especificar uma única zona de disponibilidade para todos os seus nós, permitir ElastiCache a escolha de uma única zona de disponibilidade para todos os seus nós, especificar as zonas de disponibilidade para cada nó ou permitir ElastiCache a escolha de uma zona de disponibilidade para cada nó. Novos nós podem ser criados em diferentes zonas de disponibilidade à medida que você os adiciona a um cluster do Memcached existente. Depois que um nó de cache é criado, sua zona de disponibilidade não pode ser modificada.

Se você quiser que um cluster em um único cluster de zona de disponibilidade tenha seus nós distribuídos em várias zonas de disponibilidade, ElastiCache pode criar novos nós nas várias zonas de disponibilidade. Em seguida, você pode excluir alguns ou todos os nós de cache originais. Recomendamos essa abordagem.

Para migrar nós do Memcached de uma única zona de disponibilidade para várias zonas de disponibilidade

1. Modifique seu cluster criando novos nós de cache nas zonas de disponibilidade onde você deseja. Na sua solicitação, faça o seguinte:
 - Defina AZMode (CLI: `--az-mode`) como `cross-az`.
 - Defina NumCacheNodes (CLI: `--num-cache-nodes`) como o número de nós de cache ativos no momento mais o número de novos nós de cache que você deseja criar.

- Defina `NewAvailabilityZones` (CLI: `--new-availability-zones`) como uma lista das zonas nas quais você deseja que os novos nós de cache sejam criados. Para permitir ElastiCache determinar a zona de disponibilidade para cada novo nó, não especifique uma lista.
- Defina `ApplyImmediately` (CLI: `--apply-immediately`) como verdadeiro.

 Note

Se você não estiver usando a descoberta automática, atualize sua aplicação cliente com os novos endpoints do nó de cache.

Antes de passar para a próxima etapa, certifique-se de que os nós do Memcached estão totalmente criados e disponíveis.

2. Modifique seu cluster removendo os nós que você não deseja mais na zona de disponibilidade original. Na sua solicitação, faça o seguinte:
 - Defina `NumCacheNodes` (CLI: `--num-cache-nodes`) como o número de nós de cache ativos que você deseja após esta modificação ser aplicada.
 - Defina `CacheNodeIdsToRemove` (CLI: `--nodes-to-remove`) como uma lista dos nós de cache que você deseja remover do cluster.

O número de IDs de nó de cache listados deve ser igual ao número de nós ativos no momento menos o valor em `NumCacheNodes`.
 - (Opcional) Defina `ApplyImmediately` (CLI: `--apply-immediately`) como verdadeiro.

Se você não definir `ApplyImmediately` (CLI: `--apply-immediately`) como verdadeiro, as exclusões de nó ocorrerão na sua próxima janela de manutenção.

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	

Nome da região/região	Endpoint	Protocolo	
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	
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	

Nome da região/região	Endpoint	Protocolo	
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	
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	

Nome da região/região	Endpoint	Protocolo	
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	
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	

Nome da região/região	Endpoint	Protocolo	
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	
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	

Nome da região/região	Endpoint	Protocolo	
-----------------------	----------	-----------	--

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](#).

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.

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](#)

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:

- 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 Memcached que use a sub-rede do ElastiCache na zona local.

Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).

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-

premissas 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.
- Multi-AZ (não há suporte para replicação cruzada do Outpost).
- 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 Memcached

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 Memcached.
3. Em Cluster Engine (Mecanismo do cluster), selecione Memcached.
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 Configurações do Memcached:
 - a. Name (Nome): insira um nome para o cluster do Memcached.
 - b. Description (Descrição): insira uma descrição para o cluster do Memcached.

- 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, 11211. 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. 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 nodes (Número de nós): insira o número de nós desejados no cluster.
2. Em Advanced Memcached settings (Configurações avançadas do Memcached):
- 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.
 - 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 Memcached, 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 \  
  \
```

Projeto e gerenciamento do próprio cluster do ElastiCache para implementação do Memcached

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

ElastiCache para componentes e recursos do Memcached

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

Tópicos

- [ElastiCache nós](#)
- [ElastiCache para clusters Memcached](#)
- [AWS Regiões e zonas de disponibilidade](#)
- [ElastiCache para endpoints Memcached](#)
- [ElastiCache grupos de parâmetros](#)
- [ElastiCache segurança](#)
- [ElastiCache grupos de sub-redes](#)
- [ElastiCache para eventos do Memcached](#)

ElastiCache nós

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

Um nó é um bloco de tamanho fixo da RAM em armazenamento anexado a rede seguro. Cada nó executa uma instância do Memcached. 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 Memcached](#).

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 pode comprar nós reservados a uma taxa por hora significativamente 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 vários nós reservados para serem executados na maior parte do tempo e 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](#).

O mecanismo Memcached comporta o recurso Descoberta automática. Descoberta automática é um recurso para os programas clientes identificarem automaticamente todos os nós em um cluster de cache iniciarem e manterem conexões com todos esses nós. Com a Descoberta automática, seu aplicativo não precisa se conectar manualmente com nós específicos. Na verdade, seu aplicativo conecta-se a um endpoint de configuração. A entrada DNS do endpoint de configuração contém as entradas CNAME para cada um dos endpoints do nó de cache. Portanto, ao se conectar ao endpoint de configuração, seu aplicativo recebe informações imediatamente sobre todos os nós do cluster e pode se conectar a todos eles. Você não precisa codificar endpoints de nó de cache individuais em seu aplicativo. Para obter mais informações consulte [Descoberta automática](#).

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

ElastiCache para clusters Memcached

Um cluster Memcached é um agrupamento lógico de um ou mais [ElastiCache nós](#). Os dados são particionados em todos os nós em um cluster Memcached.

Muitas ElastiCache operações são direcionadas a clusters:

- Criar um cluster
- Modificar um cluster
- 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.

Se você precisar exceder esses limites, faça sua solicitação usando o [formulário de solicitação de nós de ElastiCache cache da Amazon](#).

- [Atenuar falhas](#)

Informações sobre como melhorar a tolerância a falhas de seus clusters.

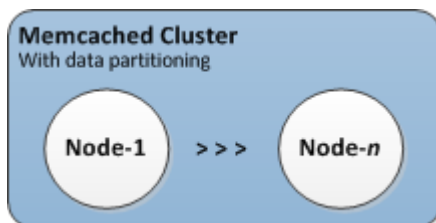
Configurações de cluster típicas

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

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

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

À medida que a demanda em seu cluster Memcached mudar, você poderá expandir ou reduzir adicionando ou removendo nós, e isso reparticiona seus dados entre o novo número de nós. Ao particionar seus dados, recomendamos usar o hashing consistente. Para obter mais informações sobre o hashing consistente, consulte [Configuração do seu cliente do ElastiCache para balanceamento de carga eficiente](#). No diagrama a seguir, você pode ver exemplos de clusters Memcached com um único nó e com vários nós.



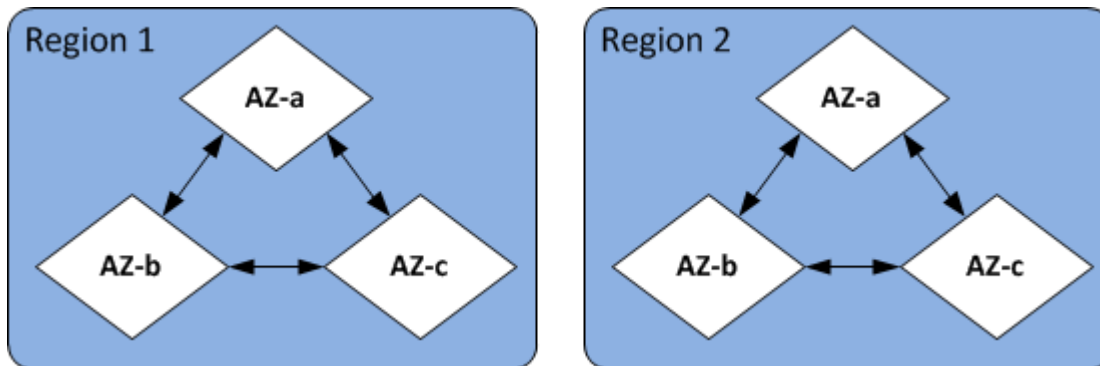
AWS Regiões e zonas de disponibilidade

O Amazon ElastiCache for Memcached 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 para uso em suas solicitações HTTP, nos AWS SDKs e no console AWS CLI. ElastiCache

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](#).



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

ElastiCache para endpoints Memcached

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

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

Para obter mais informações, consulte [Endpoints do](#) .

ElastiCache grupos de parâmetros

Parameter groups de cache são uma maneira fácil de gerenciar as configurações de tempo de execução do software do mecanismo com suporte. Os parâmetros são usados para controlar o uso de memória, as políticas de remoção, o tamanho dos itens e muito mais. Um grupo de ElastiCache parâmetros é uma coleção nomeada de parâmetros específicos do mecanismo que você pode

aplicar a um cluster. Ao fazê-lo, você garante que todos os nós do cluster sejam configurados exatamente da mesma forma.

Para obter uma lista de parâmetros com suporte, seus valores padrão e quais podem ser modificados, consulte [DescribeEngineDefaultParameters](#) ([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 segurança

Para maior segurança, o acesso aos ElastiCache nós é restrito a aplicativos executados em instâncias do Amazon EC2 incluídas na lista branca. É possível controlar as instâncias do Amazon EC2 que podem acessar seu cluster usando grupos de segurança.

Por padrão, todos os novos ElastiCache clusters são lançados em um ambiente Amazon Virtual Private Cloud (Amazon VPC). Você pode usar grupos de sub-redes para conceder acesso ao cluster a partir de instâncias do Amazon EC2 em execução em sub-redes específicas. Se você optar por executar seu cluster fora da Amazon VPC, poderá criar grupos de segurança para autorizar as instâncias do Amazon EC2 em execução dentro de grupos de segurança do Amazon EC2 específicos.

ElastiCache grupos de sub-redes

Um grupo de sub-redes é um conjunto de sub-redes (normalmente privadas) que você pode designar para seus clusters em execução em um ambiente Amazon Virtual Private Cloud (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-rede de cache em um ambiente do Amazon VPC, consulte [Segurança do Amazon VPCs e do ElastiCache](#), [Authorize access](#) e [Sub-redes e grupos de sub-redes](#).

ElastiCache para eventos do Memcached

Quando eventos significativos acontecem em um cluster de cache, ElastiCache envia uma notificação para um tópico específico do Amazon SNS. Esses eventos significativos podem incluir fatores como falha ao adicionar um nó, êxito ao adicionar um nó, modificação de um security group

etc. Ao monitorar eventos importantes, você pode se manter informado sobre o estado atual de seus clusters e, dependendo do evento, poderá executar uma ação corretiva.

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

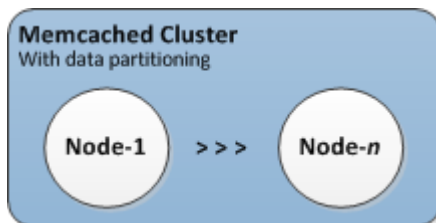
Gerenciamento de clusters

Um cluster é uma coleção de um ou mais nós de cache, todos executando uma instância do software do mecanismo de cache do Memcached. 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 Memcached. Os clusters do Memcached contêm de 1 a 60 nós nos quais você particiona horizontalmente seus dados.

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 cluster Memcached normal tem a seguinte aparência.



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 [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#).

Para obter uma lista das versões compatíveis do Memcached, consulte [Compatível com ElastiCache versões do Memcached](#).

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 Memcached versão 1.6.6 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 criar um cluster de cache e escolher pilha dupla como tipo de rede, você precisa designar um tipo de descoberta de IP: IPv4 ou IPv6. O ElastiCache usará como padrão o tipo de rede e a descoberta de IP como IPv6, mas isso pode ser alterado. Se você usar a Descoberta automática, somente os endereços IP do tipo de IP escolhido serão retornados ao cliente Memcached.

Para 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 graças à Descoberta automática, pois permite migrações (ou

reversões) de um tipo de IP de descoberta IPv4 para um tipo de IP de descoberta 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 `config get cluster` 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 de cache usando o AWS Management Console, em Connectivity (Conectividade), 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 cluster do Memcached \(console\)](#).

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 memcached \
  --cache-node-type cache.m5.large \
  --num-cache-nodes 1 \
  --network-type dual_stack \
  --ip-discovery ipv4
```

Para Windows:

```
aws elasticache create-cache-cluster ^
  --cache-cluster-id "cluster-test" ^
  --engine memcached ^
  --cache-node-type cache.m5.large ^
  --num-cache-nodes 1 ^
  --network-type dual_stack ^
  --ip-discovery ipv4
```

Identificar automaticamente os nós no seu cluster

Para clusters executando o mecanismo Memcached, o ElastiCache oferece suporte para a Descoberta automática, a capacidade de os programas clientes identificarem automaticamente todos os nós em um cluster de cache e iniciar e manter conexões com todos esses nós.

Note

A Descoberta automática é adicionada para clusters de cache executados no Memcached do Amazon ElastiCache.

Com a Descoberta automática, seu aplicativo não precisa se conectar manualmente a nós de cache individuais. Em vez disso, seu aplicativo se conecta a um nó Memcached e recupera a lista de nós. Nessa lista, seu aplicativo está ciente do restante dos nós no cluster e pode se conectar a qualquer um deles. Você não precisa embutir em código os endpoints de nó de cache individuais no seu aplicativo.

Se você estiver usando um tipo de rede de pilha dupla no cluster, a Descoberta automática só vai retornar endereços IPv4 ou IPv6, dependendo de qual deles você selecionar. Para obter mais informações, consulte [Escolher um tipo de rede](#)

Todos os nós de cache no cluster mantêm uma lista de metadados sobre todos os outros nós. Esses metadados são atualizados sempre que os nós são adicionados ou removidos do cluster.

Tópicos

- [Benefícios da descoberta automática](#)
- [Como a Descoberta automática funciona](#)
- [Como usar a Descoberta automática](#)

- [Conectar-se a nós de cache manualmente](#)
- [Adicionar a Descoberta automática à sua biblioteca de clientes](#)
- [Clientes ElastiCache com descoberta automática](#)

Benefícios da descoberta automática

A descoberta automática oferece os seguintes benefícios:

- Quando você aumenta o número de nós em um cluster de cache, os novos nós se registram no endpoint da configuração e em todos os outros nós. Quando você remove os nós do cluster de cache, os nós de partida cancelam o registro. Em ambos os casos, todos os outros nós no cluster são atualizados com os metadados do nó de cache mais recentes.
- As falhas de nós de cache são detectadas automaticamente. Os nós com falha são automaticamente substituídos.

Note

Até que a substituição dos nós seja concluída, o nó continuará a falhar.

- Um programa cliente só precisa se conectar ao endpoint da configuração. Depois disso, a biblioteca da Descoberta automática se conecta a todos os outros nós do cluster.
- Os programas cliente sondam o cluster uma vez por minuto (esse intervalo pode ser ajustado, se necessário). Se houver alguma alteração na configuração do cluster, como nós novos ou excluídos, o cliente receberá uma lista atualizada de metadados. Em seguida, o cliente se conecta ou desconecta desses nós conforme necessário.

A Descoberta automática está habilitada em todos os clusters de cache Memcached do ElastiCache. Você não precisa reinicializar nenhum dos seus nós de cache para usar esse recurso.

Como a Descoberta automática funciona

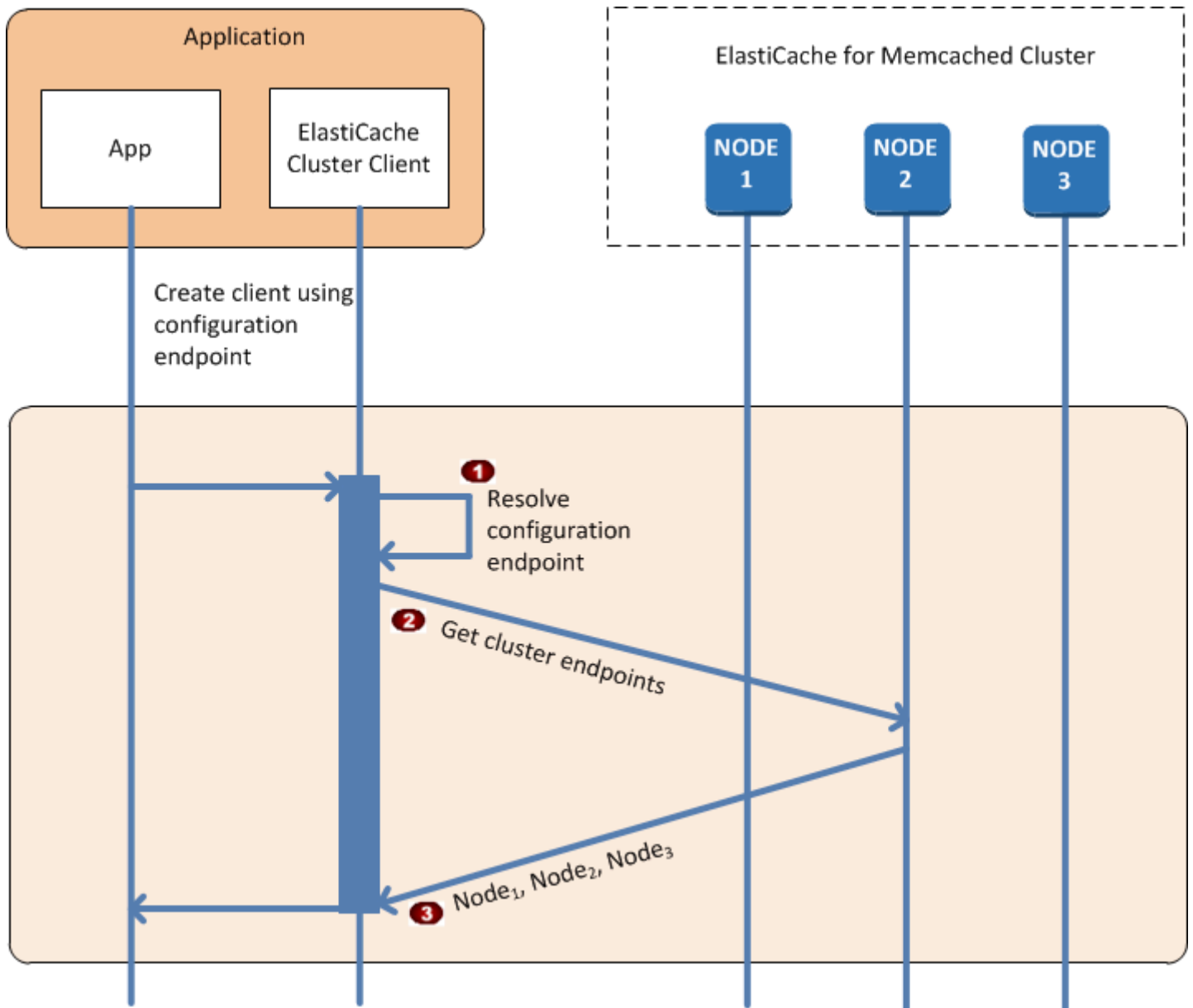
Tópicos

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

Esta seção descreve como as aplicações cliente usam o cliente de cluster do ElastiCache para gerenciar conexões de nós de cache e interagir com itens de dados no cache.

Conexão com nós de cache


Do ponto de vista do aplicativo, conectar-se ao endpoint de configuração de cluster não é diferente de conectar-se diretamente a um nó de cache individual. O diagrama de sequência a seguir mostra o processo de conexão com nós de cache.



Processo de conexão com nós de cache

- O aplicativo resolve o nome DNS do endpoint de configuração. Como o endpoint de configuração mantém entradas CNAME para todos os nós de cache, o nome DNS é resolvido para um dos nós. O cliente pode então se conectar a esse nó.
- O cliente solicita as informações de configuração para todos os outros nós. Como cada nó mantém informações de configuração para todos os nós do cluster, qualquer nó pode transmitir informações de configuração ao cliente mediante solicitação.

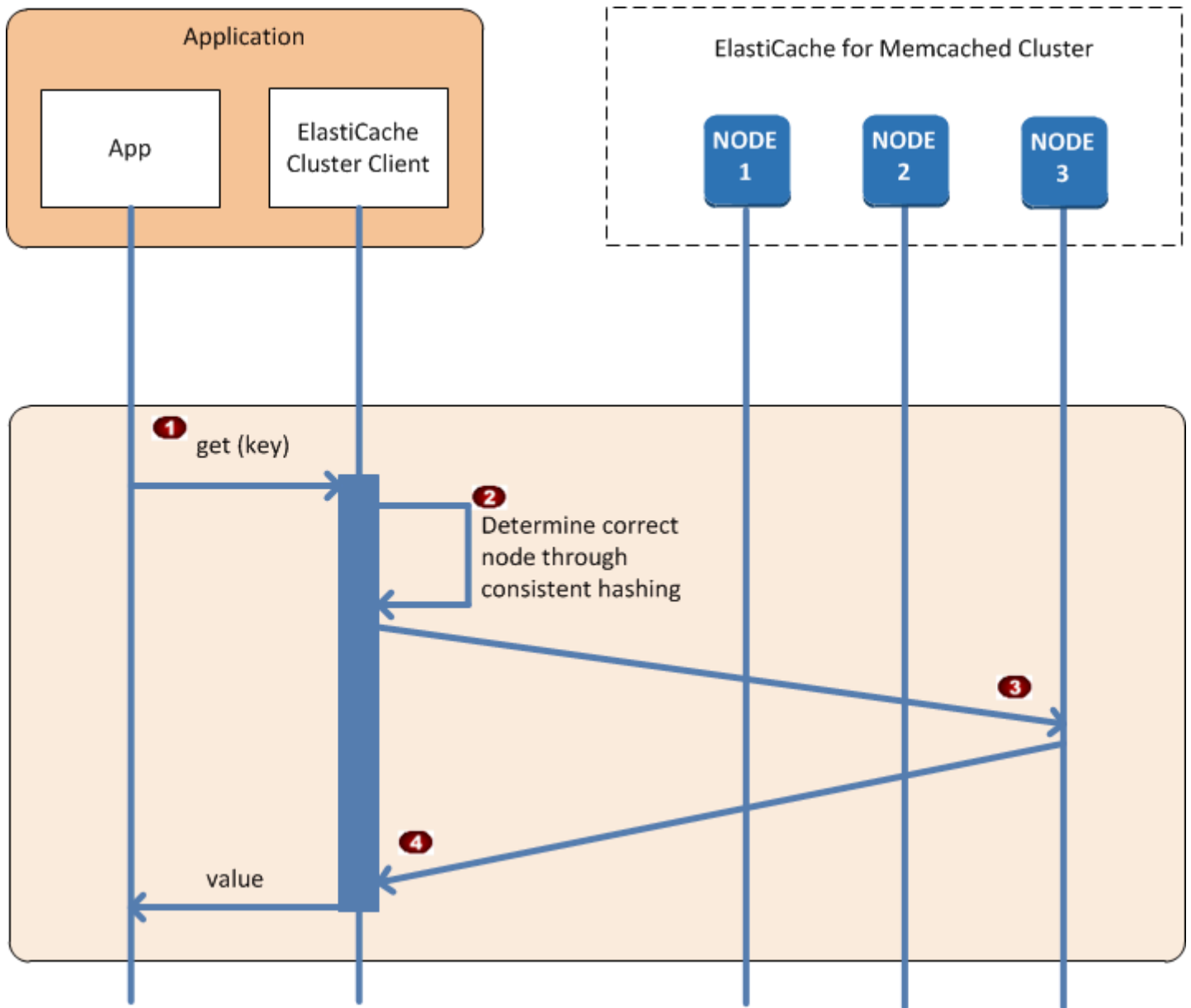
- O cliente recebe a lista atual de nomes de host e endereços IP de nós de cache. Em seguida, ele pode se conectar a todos os outros nós do cluster.

 Note

O programa cliente atualiza sua lista de nomes de host e endereços IP de nós de cache uma vez por minuto. Esse intervalo de sondagem pode ser ajustado se necessário.

Operações normais de cluster

Quando a aplicação estiver conectada a todos os nós de cache, o cliente de cluster do ElastiCache determinará quais nós deverão armazenar os itens de dados individuais e quais nós deverão ser consultados para esses itens de dados mais tarde. O diagrama de sequência a seguir mostra o processo das operações normais de cluster.



Processo das operações normais de cluster

- O aplicativo emite uma solicitação get para um item de dados específico, identificado por sua chave.
- O cliente usa um algoritmo de hash contra a chave para determinar qual nó de cache contém o item de dados.
- O item de dados é solicitado do nó apropriado.
- O item de dados é retornado ao aplicativo.

Outras operações do

Em algumas situações, você pode alterar os nós de um cluster. Por exemplo, você pode acrescentar um nó adicional para acomodar demanda adicional ou excluir um nó para economizar dinheiro durante períodos de demanda reduzida. Ou você pode substituir um nó devido a uma falha de nó de uma ou outra classificação.

Quando há uma alteração no cluster que requer uma atualização de metadados para os endpoints do cluster, essa alteração é feita em todos os nós ao mesmo tempo. Portanto, os metadados em qualquer nó especificado são consistentes com os metadados em todos os outros nós do cluster.

Em cada um desses casos, os metadados são consistentes entre todos os nós em todos os momentos, já que os metadados são atualizados ao mesmo tempo para todos os nós do cluster. Você sempre deve usar o endpoint de configuração para obter os endpoints dos vários nós no cluster. Ao usar o endpoint de configuração, você garante que não obterá dados de endpoint de um nó que "desaparecerá".

Adicionar um nó

Durante o tempo em que o nó está sendo configurado, seu endpoint não é incluído nos metadados. Assim que o nó estiver disponível, ele será adicionado aos metadados de cada um dos nós do cluster. Nesse cenário, os metadados são consistentes entre todos os nós, e você poderá interagir com o novo nó somente depois que ele estiver disponível. Antes de o nó estar disponível, você não saberá sobre ele e interagirá com os nós no seu cluster como se o novo nó não existisse.

Excluir um nó

Quando um nó é removido, seu endpoint é primeiramente removido dos metadados e, em seguida, o nó é removido do cluster. Nesse cenário, os metadados em todos os nós são consistentes e em nenhuma ocasião ele conterá o endpoint para o nó a ser removido se esse nó não estiver disponível. Durante o tempo de remoção do nó, ele não é relatado nos metadados e, portanto, seu aplicativo só interagirá com os $n-1$ nós restantes, como se o nó não existisse.

Substituir um nó

Se um nó falhar, o ElastiCache o retirará e configurará uma substituição. O processo de substituição demora alguns minutos. Durante esse tempo, os metadados em todos os nós ainda mostram o endpoint do nó com falha, mas qualquer tentativa de interação com esse nó falhará. Portanto, sua lógica sempre deve incluir uma lógica de nova tentativa.

Como usar a Descoberta automática

Para começar a usar a Descoberta automática, siga estas etapas:

- [Etapa 1: Obter o endpoint de configuração](#)
- [Etapa 2: Baixar o cliente de cluster do ElastiCache](#)
- [Etapa 3: Modificar seu programa aplicativo](#)

Etapa 1: Obter o endpoint de configuração

Para se conectar a um cluster, os programas cliente devem conhecer o endpoint de configuração de cluster. Consulte o tópico [Localização de endpoints de um cluster \(console\)](#)

Você também pode usar o comando `aws elasticache describe-cache-clusters` com o parâmetro `--show-cache-node-info`:

Seja qual for o método usado para encontrar os endpoints do cluster, o endpoint de configuração sempre terá `.cfg` em seu endereço.

Exemplo Localização de endpoints usando a AWS CLI para o ElastiCache

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

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

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {
```



```
    "CacheNodeId": "0001",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
    "CustomerAvailabilityZone": "us-east-1e"
  },
  {
    "CacheNodeId": "0002",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
    "CustomerAvailabilityZone": "us-east-1a"
  }
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "Multiple",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
"NumCacheNodes": 2,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {},
"PreferredMaintenanceWindow": "sat:06:00-sat:07:00",
```

```
        "CacheNodeType": "cache.r3.large"
    }
  ]
}
```

Etapa 2: Baixar o cliente de cluster do ElastiCache

Para tirar proveito da Descoberta automática, os programas cliente devem usar o cliente de cluster do ElastiCache. O cliente de cluster do ElastiCache está disponível para Java, PHP e .NET e contém toda a lógica necessária para descobrir e conectar-se a todos os seus nós de cache.

Para baixar o cliente de cluster do ElastiCache

1. Faça login no Console de Gerenciamento da AWS e abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.
2. No console do ElastiCache, escolha ElastiCache Cluster Client (Cliente de cluster do ElastiCache), e depois escolha Download (Baixar).

O código-fonte do cliente de cluster do ElastiCache para Java está disponível em <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java>. Essa biblioteca se baseia no popular cliente Spymemcached. O cliente de cluster do ElastiCache é lançado sob a Licença de software da Amazon <https://aws.amazon.com/asl>. Você tem a liberdade de modificar o código-fonte como você bem entender. Você pode até mesmo incorporar o código em outras bibliotecas Memcached de código aberto ou no seu próprio código de cliente.

Note

Para usar o cliente de cluster do ElastiCache para PHP, você precisa primeiro instalá-lo na sua instância do Amazon EC2. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

Para clientes compatíveis com TLS, baixe o binário com PHP versão 7.4 ou superior.

Para usar o cliente de cluster do ElastiCache para .NET, você precisa primeiro instalá-lo na sua instância do Amazon EC2. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para .NET](#).

Etapa 3: Modificar seu programa aplicativo

Modifique seu programa aplicativo para que ele use a Descoberta automática. As seções a seguir mostram como usar o cliente de cluster do ElastiCache para Java, PHP e .NET.

Important

Ao especificar o endpoint de configuração do cluster, certifique-se de que o endpoint tenha ".cfg" em seu endereço, conforme mostrado aqui. Não use um CNAME ou um endpoint sem ".cfg".

```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

Se o endpoint de configuração de cluster não for especificado explicitamente, a configuração ocorrerá para um nó específico.

Uso do cliente de cluster do ElastiCache para Java

O programa abaixo demonstra como usar o cliente de cluster do ElastiCache para se conectar a um endpoint de configuração de cluster e adicionar um item de dados ao cache. Usando a Descoberta automática, o programa se conecta a todos os nós do cluster sem qualquer intervenção adicional.

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the &AWS;-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
            new InetSocketAddress(configEndpoint,
                clusterPort));
    }
}
```

```
// The client will connect to the other cache nodes automatically.

// Store a data item for an hour.
// The client will decide which cache host will store this item.
client.set("theKey", 3600, "This is the data value");
}
}
```

Uso do cliente de cluster do ElastiCache para PHP

O programa abaixo demonstra como usar o cliente de cluster do ElastiCache para se conectar a um endpoint de configuração de cluster e adicionar um item de dados ao cache. Usando a Descoberta automática, o programa se conectará a todos os nós do cluster sem qualquer intervenção adicional.

Para usar o cliente de cluster do ElastiCache para PHP, você precisa primeiro instalá-lo na sua instância do Amazon EC2. Para obter mais informações, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

```
<?php

/**
 * Sample PHP code to show how to integrate with the Amazon ElastiCache
 * Auto Discovery feature.
 */

/* Configuration endpoint to use to initialize memcached client.
 * This is only an example. */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the ElastiCache cluster.
 * This is only an example */
$server_port = 11211;

/**
 * The following will initialize a Memcached client to utilize the Auto Discovery
 * feature.
 *
 * By configuring the client with the Dynamic client mode with single endpoint, the
 * client will periodically use the configuration endpoint to retrieve the current
 * cache
 * cluster configuration. This allows scaling the cache cluster up or down in number
 * of nodes
 * without requiring any changes to the PHP application.
```

```
*
* By default the Memcached instances are destroyed at the end of the request.
* To create an instance that persists between requests,
*   use persistent_id to specify a unique ID for the instance.
* All instances created with the same persistent_id will share the same connection.
* See http://php.net/manual/en/memcached.construct.php for more information.
*/
$dynamic_client = new Memcached('persistent-id');
$dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
$dynamic_client->addServer($server_endpoint, $server_port);

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

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

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

Para obter um exemplo de como usar o cliente do ElastiCache Cluster com TLS habilitado, consulte [Using in transit encryption with PHP and Memcached](#).

Uso do cliente de cluster do ElastiCache para .NET

Note

O cliente de cluster ElastiCache .NET foi suspenso em maio de 2022.

O cliente .NET para o ElastiCache é um código aberto que pode se encontrado em <https://github.com/awslabs/elasticache-cluster-config-net>.

Em geral, os aplicativos .NET obtém suas configurações de seu arquivo de configuração. O seguinte é um exemplo de arquivo de configuração de aplicativo.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section
      name="clusterclient"
      type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
Amazon.ElastiCacheCluster" />
  </configSections>

  <clusterclient>
    <!-- the hostname and port values are from step 1 above -->
    <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
port="11211" />
  </clusterclient>
</configuration>
```

O programa em C# abaixo demonstra como usar o cliente de cluster do ElastiCache para se conectar a um endpoint de configuração de cluster e adicionar um item de dados ao cache. Usando a Descoberta automática, o programa se conectará a todos os nós do cluster sem qualquer intervenção adicional.

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

using System;

using Amazon.ElastiCacheCluster;
```

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

public class DotNetAutoDiscoveryDemo {

    public static void Main(String[] args) {

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

        // Store the data for 3600 seconds (1hour) in the cluster.
        // The client will decide which cache host will store this item.
        memClient.Store(StoreMode.Set, 3600, "This is the data value.");

    } // end Main

} // end class DotNetAutoDiscoverDemo
```

Conectar-se a nós de cache manualmente

Se o seu programa cliente não usar a Descoberta automática, ele poderá se conectar manualmente a cada um dos nós de cache. Este é o comportamento padrão para clientes Memcached.

Você pode obter uma lista de nomes de host de nós de cache e números de portas no [Console de Gerenciamento da AWS](#). Você também pode usar o comando AWS CLI `aws elasticache describe-cache-clusters` com o parâmetro `--show-cache-node-info`.

Example

O seguinte trecho de código Java mostra como se conectar a todos os nós em um cluster de cache de quatro nós:

```
...  
  
ArrayList<String> cacheNodes = new ArrayList<String>(Arrays.asList(  
    "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",  
    "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",  
    "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",  
    "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));  
  
MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));  
  
...
```

Important

Se você expandir ou reduzir seu cluster de cache adicionando ou removendo nós, precisará atualizar a lista de nós no código do cliente.

Adicionar a Descoberta automática à sua biblioteca de clientes

As informações de configuração para Descoberta automática são armazenadas de forma redundante em cada nó de cluster de cache. Os aplicativos cliente podem consultar qualquer nó de cache e obter as informações de configuração para todos os nós no cluster.

A maneira como um aplicativo faz isso depende da versão do mecanismo de cache:

- Se a versão do mecanismo de cache for 1.4.14 ou superior, use o comando `config`.
- Se a versão do mecanismo de cache for inferior a 1.4.14, use o comando `get AmazonElastiCache:cluster`.

As saídas desses dois comandos são idênticas e estão descritas na seção [Output Format](#) abaixo.

Versão do mecanismo de cache 1.4.14 ou superior

Para o mecanismo de cache versão 1.4.14 ou superior, use o comando `config`. Esse comando foi adicionado ao protocolos ASCII e binários do Memcached pelo ElastiCache, e é implementado no cliente de cluster do ElastiCache. Se você quiser usar a Descoberta automática com outra biblioteca de cliente, essa biblioteca precisará ser estendida para oferecer suporte ao comando `config`.

Note

A seguinte documentação pertence ao protocolo ASCII. No entanto, o comando `config` oferece suporte para ASCII e binário. Se você deseja adicionar suporte para Descoberta automática usando o protocolo binário, consulte o [código-fonte do cliente de cluster do ElastiCache](#).

Sintaxe

```
config [sub-command] [key]
```

Opções

Name (Nome)	Descrição	Obrigatório
sub-command		Sim

Name (Nome)	Descrição	Obrigatório
	O subcomando usado para interagir com um nó de cache. Para a Descoberta automática, esse subcomando é <code>get</code> .	
<code>key</code>	A chave na qual a configuração de cluster está armazenada. Para a Descoberta automática, essa chave se chama <code>cluster</code> .	Sim

Para obter as informações de configuração do cluster, use o seguinte comando:

```
config get cluster
```

Versão do mecanismo de cache inferior a 1.4.14

Para obter as informações de configuração do cluster, use o seguinte comando:

```
get AmazonElastiCache:cluster
```

Note

Não manipule a chave "AmazonElastiCache: cluster", uma vez que ela é onde as informações de configuração de cluster residem. Se você substituir essa chave, o cliente poderá ser incorretamente configurado por um curto período (não mais de 15 segundos) antes que o ElastiCache atualize automaticamente e corretamente as informações de configuração.

Output Format

Se você usar `config get cluster` ou `get AmazonElastiCache:cluster`, a resposta consistirá em duas linhas:

- O número de versão das informações de configuração. Cada vez que um nó é adicionado ou removido do cluster de cache, o número da versão aumenta em um.
- Uma lista de nós de cache. Cada nó na lista é representado por um grupo de nome do host| endereço IP|porta, e cada nó é delimitado por um espaço.

Um retorno de carro e um caractere de avanço de linha (CR + LF) aparecem no final de cada linha. A linha de dados contém um caractere de avanço de linha (LF) no final, ao qual o CR+LF é adicionado. A linha de versão de configuração é encerrada por LF sem o CR.

Um cluster de cache contendo três nós seria representado da seguinte maneira:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Cada nó é mostrado com o CNAME e o endereço IP privado. O CNAME sempre estará presente. Se o endereço IP privado não estiver disponível, ele não será mostrado. Entretanto, os caracteres de barra vertical "|" continuarão a ser impressos.

Example

Veja a seguir um exemplo da carga útil retornada quando você consulta as informações de configuração:

```
CONFIG cluster 0 136\r\n
12\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
  myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\n
END\r\n
```

Note

- A segunda linha indica que as informações de configuração foram modificadas doze vezes até agora.
- Na terceira linha, a lista de nós está em ordem alfabética por nome de host. Essa ordem pode estar em uma sequência diferente da que você está usando no seu aplicativo cliente.

Cientes ElastiCache com descoberta automática

Esta seção discute a instalação e a configuração de clientes PHP e .NET do ElastiCache.

Tópicos

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

Instalação e compilação de clientes de cluster

Esta seção discute a instalação, a configuração e a compilação dos clientes de cluster de descoberta automática PHP e .NET do Amazon ElastiCache.

Tópicos

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

Instalação do cliente de cluster do ElastiCache para .NET

Note

O cliente de cluster ElastiCache .NET foi suspenso em maio de 2022.

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

Esta seção descreve como instalar, atualizar e remover os componentes .NET do cliente de cluster do ElastiCache em instâncias do Amazon EC2. Para obter mais informações sobre a descoberta automática, consulte [Auto discovery](#). Para obter um código .NET de exemplo para usar o cliente, consulte [Auto discovery with DotNET](#).

Tópicos

- [Instalar o .NET](#)
- [Baixar o cliente de cluster .NET do ElastiCache para ElastiCache](#)
- [Instalar conjuntos da AWS com o NuGet](#)

Instalar o .NET

Você deve ter o .NET 3.5 ou posterior instalado para usar o SDK .NET da AWS para ElastiCache. Caso não tenha o .NET 3.5 ou posterior, pode fazer download e instalar a versão mais recente em <http://www.microsoft.com/net>.

Baixar o cliente de cluster .NET do ElastiCache para ElastiCache

Para baixar o cliente de cluster .NET do ElastiCache

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

Instalar conjuntos da AWS com o NuGet

O NuGet é um sistema de gerenciamento de pacotes para a plataforma .NET. O NuGet está ciente das dependências de assembly e instala todos os arquivos necessários automaticamente. Os assemblies instalados pelo NuGet são armazenados com a sua solução, em vez de em uma localização central, como em Program Files, e, por isso, você pode instalar versões específicas para um aplicativo sem criar problemas de compatibilidade.

Instalar o NuGet

O NuGet pode ser instalado da galeria de instalação no MSDN; consulte <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Se você estiver usando o Visual Studio 2010 ou posterior, o NuGet será instalado automaticamente.

Você pode usar o NuGet no Gerenciador de Soluções ou no Console do Gerenciador de Pacotes.

Usar o NuGet no Gerenciador de Soluções

Para usar o NuGet no Gerenciador de Soluções do Visual Studio 2010

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

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

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

Na linha de comando, você pode instalar os assemblies usando Install-Package, conforme mostrado a seguir.

```
Install-Package Amazon.ElastiCacheCluster
```

Para ver uma página de cada pacote disponível via NuGet, como os conjuntos SDK da AWS e extensões da AWS, consulte o site do NuGet em <http://www.nuget.org>. A página de cada pacote inclui uma linha de comando de amostra para instalar o pacote usando o console e uma lista das versões anteriores desse pacote que estão disponíveis no NuGet.

Para obter mais informações sobre os comandos do Console do Gerenciador de Pacotes, consulte <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>.

Instalação do cliente de cluster do ElastiCache para PHP

Esta seção descreve como instalar, atualizar e remover os componentes de PHP do cliente de cluster do ElastiCache em instâncias do Amazon EC2. Para obter mais informações sobre a descoberta automática, consulte [Identificar automaticamente os nós no seu cluster](#). Para obter o código de exemplo do PHP para usar o cliente, consulte [Uso do cliente de cluster do ElastiCache para PHP](#).

Tópicos

- [Para baixar o pacote de instalação](#)
- [Para usuários que já possuam a extensão php-memcached instalada](#)
- [Etapas de instalação para novos usuários](#)
- [Remoção do cliente de cluster PHP](#)

Para baixar o pacote de instalação

Para garantir que está usando a versão correta do cliente de cluster do ElastiCache para PHP, você precisa saber qual versão do PHP está instalada na sua instância do Amazon EC2. Você também precisará saber se a sua instância do Amazon EC2 está executando uma versão de 64 bits ou 32 bits do Linux.

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

- No prompt de comando, execute o seguinte comando:

```
php -v
```

A versão do PHP será mostrada na saída, como neste exemplo:

```
PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
```

Note

Se suas versões do PHP e do Memcached forem incompatíveis, você receberá uma mensagem de erro como a seguinte:

```
PHP Warning: PHP Startup: memcached: Unable to initialize module
Module compiled with module API=20100525
PHP compiled with module API=20131226
These options need to match
in Unknown on line 0
```

Se isso acontecer, será necessário compilar o módulo do código-fonte. Para obter mais informações, consulte [Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP](#).

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

1. Faça login no AWS Management Console e abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. Na lista Instances (Instâncias), clique na sua instância do Amazon EC2.
3. Na guia Description, procure o campo AMI:. Uma instância de 64 bits deve ter x86_64 como parte da descrição. Para uma instância de 32 bits, procure i386 ou i686 neste campo.

Agora, você está pronto para baixar o cliente de cluster do ElastiCache.

Para baixar o cliente de cluster do ElastiCache para PHP

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

Para clientes compatíveis com TLS, baixe o binário com PHP versão 7.4 ou superior.

Para usuários que já possuam a extensão php-memcached instalada

Para atualizar a instalação do **php-memcached**

1. Remova a instalação anterior da extensão Memcached para PHP, conforme descrito pelo tópico [Remoção do cliente de cluster PHP](#).
2. Instale a nova extensão php-memcached do ElastiCache conforme descrito anteriormente em [Etapas de instalação para novos usuários](#).

Etapas de instalação para novos usuários

Tópicos


- [Instalação do PHP 7.x - 8.x para novos usuários](#)
- [Instalação do PHP 5.x para novos usuários](#)

Instalação do PHP 7.x - 8.x para novos usuários

Tópicos

- [Para instalar o PHP 7.x - 8.x em uma AMI do Amazon Linux 2](#)
- [Para instalar o PHP 7.x - 8.x em uma AMI 201609 do Amazon Linux](#)
- [Para instalar o PHP 7.x - 8.x em uma AMI do SUSE Linux 15](#)
- [Para instalar o PHP 7.x - 8.x em uma AMI do Ubuntu 22.04](#)

Para instalar o PHP 7.x - 8.x em uma AMI do Amazon Linux 2

 Note

Se necessário, substitua o *PHP-7.x* pela versão que você estiver usando.

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

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

3. Instale o PHP 7.x usando `amazon-linux-extras`

Com o Amazon Linux 2, você pode usar a Biblioteca de extras para instalar atualizações de aplicação e software em suas instâncias. Essas atualizações de software são conhecidas como tópicos. É possível instalar uma versão específica de um tópico ou omitir informações de versão para usar a mais recente. Para obter mais informações, consulte [Biblioteca de extras \(Amazon Linux 2\)](#).

Para isso, siga as etapas a seguir;

- a. Primeiro, verifique se amazon-linux-extras já está instalado.
- b. Se ainda não estiver instalado, execute o comando a seguir para instalá-lo:

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

- c. Confirme se o tópico do PHP 7.x está disponível na máquina do Amazon Linux 2:

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

- d. Na saída, revise todos os tópicos do PHP 7 e selecione a versão que deseja:

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

- e. Instale os pacotes de PHP dos repositórios. Por exemplo:

```
sudo yum clean metadata
```

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

4. Baixe o cliente de cluster do Amazon ElastiCache.

- Abra o console do ElastiCache em <https://console.aws.amazon.com/elasticache/>.

No painel do ElastiCache, acesse ElastiCache Cluster Client (Cliente cluster do ElastiCache) e, em seguida, escolha a versão do PHP7 que você deseja.

- Na linha de comando, substitua o PHP-7.X pela versão do PHP desejada e substitua a ARCH pela arquitetura desejada (X86 ou arm). Para o PHP >= 7.4, substitua o OpenSSL pela versão desejada do OpenSSL (openssl1.1 ou openssl3). Se você estiver usando o PHP > 7.4, remova o sufixo OpenSSL.

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

5. Use `tar -zxvf` para extrair o arquivo baixado.

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

- Com permissões de root, copie o arquivo de artefato extraído `amazon-elasticache-cluster-client.so` para `/usr/lib64/php/modules`.

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

- Adicione `extension=amazon-elasticache-cluster-client.so` ao arquivo `/etc/php.ini`
- Caso você tenha baixado o cliente de cluster do ElastiCache com PHP 7.4 ou superior, instale o OpenSSL 1.1.x ou superior. Instruções de instalação do OpenSSL 1.1.1:

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

Para instalar o PHP 7.x - 8.x em uma AMI 201609 do Amazon Linux

Note

Se necessário, substitua o `php7.x` pela versão que você estiver usando.

- Execute uma nova instância a partir da AMI. Para obter mais informações, consulte [Etapa 1: iniciar uma instância](#) no Guia do usuário do Amazon EC2.
- Execute o seguinte comando :

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

- Instale o PHP

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

4. Baixe o cliente de cluster do Amazon ElastiCache.

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

5. EXTRACT latest-64bit.

```
tar -zxvf latest-64bit
```

6. Com permissão de root, copie o arquivo de artefato extraído `amazon-elasticache-cluster-client.so` para `/usr/lib64/php/7.x/modules/`.

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

7. Crie o arquivo `50-memcached.ini`.

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

8. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 7.x - 8.x em uma AMI do SUSE Linux 15

Note

Se necessário, substitua o *php7.x* pela versão que você estiver usando.

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

```
sudo zypper refresh  
sudo zypper update -y
```

```
sudo zypper install gcc
```

3. Instale o PHP

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

or

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

4. Baixe o cliente de cluster do Amazon ElastiCache e substitua <ARCH> pela arquitetura desejada (X86 ou arm). O SUSE 15 vem com o OpenSSL1.1 integrado. Portanto, para o PHP >= 7.4, escolha o binário do cliente com OpenSSL1. Se você estiver usando o PHP < 7.4, remova o sufixo OpenSSL.

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

5. EXTRACT latest-64bit.

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

6. Com permissão de root, copie o arquivo de artefato extraído `amazon-elasticache-cluster-client.so` para `/usr/lib64/php7/extensions/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Insira a linha `extension=amazon-elasticache-cluster-client.so` no arquivo `/etc/php7/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php7/cli/php.ini
```

8. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 7.x - 8.x em uma AMI do Ubuntu 22.04

Note

Se necessário, substitua o *php7.x* pela versão que você estiver usando.

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

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

3. Instale o PHP

- a. Instruções de instalação do PHP 8.1:

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

- b. Instruções de instalação do PHP 7.4:

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

4. Baixe o cliente de cluster do Amazon ElastiCache e substitua <ARCH> pela arquitetura desejada (X86 ou arm). O Ubuntu 22.04 vem com o OpenSSL3 integrado. Portanto, para o PHP >= 7.4, escolha o binário do cliente com OpenSSL3. Se você estiver usando o PHP < 7.4, remova o sufixo OpenSSL.

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

5. Extraia a versão mais recente de 64 bits.

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

6. Com permissão de root, copie o arquivo de artefato extraído `amazon-elasticache-cluster-client.so` para o diretório de extensão `php /usr/lib/php/20190902`. Caso

esse diretório de extensão não exista, será possível encontrá-lo executando: `php -i | grep extension_dir`

7. Insira a linha `extension=amazon-elasticache-cluster-client.so` no arquivo `/etc/php/7.x/cli/php.ini`.

Instalação do PHP 5.x para novos usuários

Tópicos

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

Para instalar o PHP 5 em uma AMI Amazon Linux 2014.03 (64 bits e 32 bits)

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

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

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

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

Veja a seguir um exemplo de comando de instalação para o PHP 5.4, Linux de 64 bits. Nessa amostra, substitua `X.Y.Z` pelo número de versão atual:

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

Note

Use a versão recente do artefato de instalação.

- Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php.d` e insira `extension=amazon-elasticache-cluster-client.so` no arquivo:

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

- Inicie ou reinicie seu servidor Apache.

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

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

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

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

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

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

- Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php.d` e insira `extension=amazon-elasticache-cluster-client.so` no arquivo.

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

- Inicie ou reinicie seu servidor Apache.


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

Para instalar o PHP 5 em uma AMI Ubuntu Server 14.04 LTS (64 bits e 32 bits)

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

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

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

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

Note

Essa etapa de instalação instala o artefato de build `amazon-elasticache-cluster-client.so` no diretório `/usr/lib/php5/20121212*`. Verifique o caminho absoluto do artefato de build, porque você precisa dele na próxima etapa.

Se o comando anterior não funcionar, você precisará extrair manualmente o artefato de cliente PHP `amazon-elasticache-cluster-client.so` do arquivo `*.tgz` obtido por download e copiá-lo para o diretório `/usr/lib/php5/20121212*`.

```
$ tar -xvf <package download path>  
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php5/cli/conf.d` e insira `"extension=<absolute path to amazon-elasticache-cluster-client.so>"` no arquivo.

```
$ echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee --append /etc/php5/cli/conf.d/memcached.ini
```

6. Inicie ou reinicie seu servidor Apache.

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

Para instalar o PHP 5 para a AMI SUSE Linux Enterprise Server 11 (64 bits ou 32 bits)

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

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

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

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

5. Com a permissão root/sudo, adicione um novo arquivo denominado `memcached.ini` no diretório `/etc/php5/conf.d` e insira `extension=amazon-elasticache-cluster-client.so` no arquivo.

```
$ echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php5/conf.d/memcached.ini
```

6. Inicie ou reinicie seu servidor Apache.

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

Note

Se a Etapa 5 não funcionar para nenhuma das plataformas anteriores, verifique o caminho de instalação para `amazon-elasticache-cluster-client.so`. Além disso, especifique o caminho completo do binário da extensão. Além disso, verifique se o PHP em uso é uma versão com suporte. Oferecemos suporte às versões 5.3 a 5.5.

Outras distribuições do Linux

Em alguns sistemas, especialmente o CentOS7 e o Red Hat Enterprise Linux (RHEL) 7.1, `libsasl2.so.3` substituiu `libsasl2.so.2`. Nesses sistemas, quando você carrega o cliente de cluster ElastiCache, ele tenta e não consegue localizar e carregar `libsasl2.so.2`. Para resolver esse problema, crie um link simbólico para `libsasl2.so.3`, de forma que, quando o cliente tentar carregar `libsasl2.so.2`, ele seja redirecionado para `libsasl2.so.3`. O código a seguir cria esse link simbólico.

```
cd /usr/lib64
$ sudo ln libsasl2.so.3 libsasl2.so.2
```

Remoção do cliente de cluster PHP

Tópicos

- [Remover uma versão anterior do PHP 7 ou superior](#)
- [Remover uma versão anterior do PHP 5](#)

Remover uma versão anterior do PHP 7 ou superior

Para remover uma versão anterior do PHP 7 ou superior

1. Remova o arquivo `amazon-elasticache-cluster-client.so` do diretório lib PHP apropriado, conforme indicado anteriormente nas instruções de instalação. Consulte a seção para a sua instalação em [Para usuários que já possuam a extensão php-memcached instalada](#).
2. Remova a linha `extension=amazon-elasticache-cluster-client.so` do arquivo `php.ini`.
3. Inicie ou reinicie seu servidor Apache.

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

Remover uma versão anterior do PHP 5

Para remover uma versão anterior do PHP 5

1. Remova a extensão php-memcached:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Remova o arquivo `memcached.ini` adicionado no diretório apropriado, conforme indicado nas etapas de instalação anteriores.

Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP

Esta seção discute como obter e compilar o código-fonte do cliente de cluster do ElastiCache para PHP.

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

Tópicos

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

Compilação da biblioteca libmemcached

Bibliotecas de pré-requisitos

- OpenSSL 1.1.0 ou superior, a menos que o suporte a TLS tenha sido desabilitado pelo `./configure --disable-tls`.
- SASL (libsasl2), a menos que o suporte a SASL tenha sido desabilitado pelo `./configure --disable-sasl`.

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

1. Iniciar uma instância do Amazon EC2.
2. Instale as dependências da biblioteca.
 - Na AMI 201509 do Amazon Linux/AMI do Amazon Linux 2

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

- Na AMI do Ubuntu 14.04 (não é necessário para versões do Ubuntu que vêm com OpenSSL >= 1.1)

```
sudo apt-get update

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

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

3. Extraia o repositório e compile o código.

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

Se, ao executar `../configure`, não for possível encontrar `libssl` (biblioteca OpenSSL), poderá ser necessário ajustar a variável de ambiente `PKG_CONFIG_PATH`:

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

Como alternativa, se você não estiver usando o TLS, poderá desativá-lo executando:

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

Compilação do cliente de descoberta automática do Memcached do ElastiCache para PHP

As seções a seguir descrevem como compilar o cliente de descoberta automática do Memcached do ElastiCache

Tópicos

- [Compilar o cliente do Memcached do ElastiCache para PHP 7](#)
- [Compilação do cliente do Memcached do ElastiCache para PHP 5](#)

Compilar o cliente do Memcached do ElastiCache para PHP 7

Substitua PHP-7.x pela versão que você estiver usando.

Instale o PHP:

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

Execute o seguinte conjunto de comandos no diretório de código.

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

Se, ao executar `../configure`, não for possível encontrar a `libssl` (biblioteca OpenSSL), poderá ser necessário ajustar a variável de ambiente `PKG_CONFIG_PATH` para o diretório de arquivos `.PC` do OpenSSL:

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

Como alternativa, se você não estiver usando o TLS, poderá desativá-lo executando:

```
make
```

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

Note

Você pode vincular estaticamente a biblioteca libmemcached ao binário PHP para que ele possa ser compatibilizado em várias plataformas Linux. Para fazer isso, execute o seguinte comando antes de make:

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -
lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

Compilação do cliente do Memcached do ElastiCache para PHP 5

Compile o `aws-elasticache-cluster-client-memcached-for-php` executando os seguintes comandos na pasta `aws-elasticache-cluster-client-memcached-for-php/`.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-
php/tree/php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
make install
```


Configuração do ElastiCache

Um cluster do ElastiCache tem conformidade de protocolo com Memcached. O código, os aplicativos e as ferramentas mais populares que você usa hoje com seu ambiente Memcached existente funcionarão perfeitamente com o serviço.

Esta seção discute considerações específicas para se conectar a nós de cache no ElastiCache.

Tópicos

- [Localização de endpoints de nó e números de porta](#)
- [Conexão para uso da descoberta automática](#)
- [Nomes de DNS e IP subjacente](#)

Localização de endpoints de nó e números de porta

Para se conectar a um nó de cache, seu aplicativo precisa conhecer o endpoint e o número da porta desse nó.

Localização de endpoints de nó e números de porta (console)

Para determinar os endpoints do nó e os números de porta

1. Faça login no [Console de Gerenciamento do Amazon ElastiCache](#) e escolha o mecanismo em execução no seu cluster.

É exibida uma lista de todos os clusters que executam o mecanismo escolhido.

2. Continue abaixo para o mecanismo e a configuração que você está executando.
3. Escolha o nome do cluster de interesse.
4. Localize as colunas Port e Endpoint para o nó em que você está interessado.

Localização de endpoints de nó de cache e números de porta (AWS CLI)

Para determinar endpoints de nó de cache e números de porta, use o comando `describe-cache-clusters` com o parâmetro `--show-cache-node-info`.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Os nomes DNS e números de porta totalmente qualificados estão na seção Endpoint da saída.

Localização de endpoints de nó de cache e números de porta (API do ElastiCache)

Para determinar endpoints de nó de cache e números de porta, use a ação `DescribeCacheClusters` com o parâmetro `ShowCacheNodeInfo=true`.

Example

```
https://elasticache.us-west-2.amazonaws.com /
?Action=DescribeCacheClusters
&ShowCacheNodeInfo=true
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140421T220302Z
&Version=2014-09-30
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20140421T220302Z
&X-Amz-Expires=20140421T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

Conexão para uso da descoberta automática

Se os seus aplicativos usam a Descoberta automática, você só precisa conhecer o endpoint de configuração do cluster, e não os endpoints individuais de cada nó de cache. Para obter mais informações, consulte [Identificar automaticamente os nós no seu cluster](#).

Note

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

Nomes de DNS e IP subjacente

Os clientes mantêm uma lista de servidores contendo os endereços e as portas dos servidores que possuem os dados do cache. Ao usar o ElastiCache, a API `DescribeCacheClusters` (ou o utilitário de linha de comando `describe-cache-clusters`) retorna uma entrada DNS totalmente qualificada e um número de porta que podem ser usados para a lista de servidores.

Important

É importante que os aplicativos do cliente estejam configurados para resolver frequentemente os nomes DNS dos nós de cache quando eles tentam se conectar a um endpoint do nó de cache.

Instalações de VPC

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

Instalações não VPC

O ElastiCache garante que o nome DNS de um nó de cache permaneça inalterado quando nós de cache são recuperados em caso de falha. No entanto, o endereço IP subjacente do nó de cache pode mudar.

A maioria das bibliotecas de cliente oferece suporte para conexões persistentes de nó de cache por padrão. Recomendamos o uso de conexões de nó de cache persistentes ao usar o ElastiCache. O armazenamento em cache de DNS no lado do cliente pode ocorrer em vários locais, incluindo bibliotecas de clientes, runtime da linguagem ou sistema operacional do cliente. Você deve revisar a configuração do seu aplicativo em cada camada para garantir que esteja resolvendo os endereços IP frequentemente para seus nós de cache.

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, 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:

Você quer usar um serviço ElastiCache sem servidor ou baseado em instâncias?

Se você quiser usar o cache sem servidor, basta garantir que você tenha configurado devidamente a VPC, as sub-redes e os grupos de segurança. Para obter mais detalhes, consulte [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#). Se você quiser usar com base em instâncias ElastiCache, continue lendo.

- 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 do Memcached](#).

- 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 [Padrões de acesso para acessar um cache do ElastiCache em um Amazon VPC](#).

- 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](#).

- 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 Memcached](#)
- [Requisitos de escalabilidade](#)
- [Requisitos de acesso](#)
- [Requisitos de região, zona de disponibilidade e zona local](#)

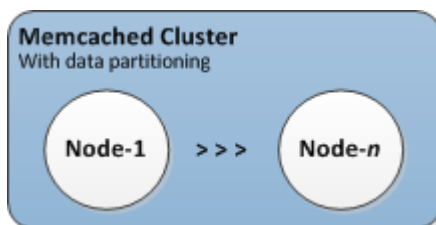
Requisitos de memória e processador

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.

O mecanismo Memcached é multi-threaded e, portanto, o número de núcleos de um nó afeta o poder computacional disponível para o cluster.

Configuração do cluster do Memcached

ElastiCache para Memcached, os clusters são compostos de 1 a 60 nós. Os dados em um cluster Memcached são particionados nos nós do cluster. Seu aplicativo conecta-se a um cluster Memcached usando um endereço de rede chamado de Endpoint. Cada nó em um cluster Memcached tem seu próprio endpoint, que seu aplicativo usa para ler ou gravar de/em um nó específico. Além dos pontos de extremidade do nó, o cluster do Memcached em si tem um endpoint chamado endpoint de configuração. Sua aplicação pode usar esse endpoint para ler ou gravar no cluster, deixando a determinação de qual nó deve ser lido ou gravado para até [Identificar automaticamente os nós no seu cluster](#).



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 Memcached, o novo cluster começa vazio.

Os clusters do Amazon ElastiCache for Memcached podem ser ampliados ou ampliados. Para expandir ou reduzir um cluster Memcached, basta adicionar ou remover nós do cluster. Se você tiver habilitado a Descoberta automática e seu aplicativo estiver se conectando ao endpoint de configuração do cluster, não será necessário fazer alterações no aplicativo quando você adicionar ou remover nós.

Para obter mais informações, consulte [Dimensionamento ElastiCache para Memcached](#) 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 [Acesso ao seu 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:

- [Regiões e zonas de disponibilidade](#)
- [Zonas locais](#)
- [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 do Memcached

Os clusters Memcached contêm um ou mais nós com os dados do cluster particionados entre os nós. Por isso, as necessidades de memória do cluster e de memória de um nó estão relacionadas, mas não são idênticas. Você pode obter a capacidade de memória de cluster necessária tendo alguns nós grandes ou vários nós menores. Além disso, conforme suas necessidades mudarem, você poderá adicionar ou remover nós do cluster e, assim, pagar apenas pelo que precisa.

A capacidade de memória total do cluster é calculada multiplicando o número de nós no cluster pela capacidade de RAM de cada nó, depois de deduzir as despesas gerais do sistema. A capacidade de cada nó é baseada no tipo de nó.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

O número de nós no cluster é um fator chave na disponibilidade do seu cluster executando o Memcached. A falha de um único nó pode ter um impacto na disponibilidade da sua aplicação e na carga do seu banco de dados de backend. Nesse caso, o ElastiCache provisiona uma substituição para um nó com falha e ele é preenchido novamente. Para reduzir esse impacto na disponibilidade, espalhe sua memória e capacidade de computação ao redor de um número maior de nós com menor capacidade, em vez de usar um número menor de nós de alta capacidade.

Em um cenário em que você deseja ter 35 GB de memória cache, você pode definir qualquer uma das seguintes configurações:

- 11 nós cache .t2.medium com 3,22 GB de memória e 2 threads cada = 35,42 GB e 22 threads.
- 6 nós cache .m4.large com 6,42 GB de memória e 2 threads cada = 38,52 GB e 12 threads.
- 3 nós cache .r4.large com 12,3 GB de memória e 2 threads cada = 36,90 GB e 6 threads.
- 3 nós cache .m4.xlarge com 14,28 GB de memória e 4 threads cada = 42,84 GB e 12 threads.

Comparar opções de nós

Tipo de nó	Memória (em GB)	Núcleos	Custo horário*	Nós necessários	Memória total (em GB)	Total de núcleos	Custo mensal
cache.t2.medium	3.22	2	0,068 USD	11	35.42	22	US\$ 538,56
cache.m4.large	6.42	2	0,156 USD	6	38.52	12	US\$ 673,92
cache.m4.xlarge	14.28	4	0,311 USD	3	42.84	12	671,76 USD
cache.m5.xlarge	12.93	4	0,311 USD	3	38.81	12	671,76 USD
cache.m6g.large	6.85	2	US\$ 0,147	6	41.1	12	US\$ 635
cache.r4.large	12.3	2	0,228 USD	3	36.9	6	492,48 USD
cache.r5.large	13.07	2	US\$ 0,216	3	39.22	6	US\$ 466,56
cache.r6g.large	13.07	2	US\$ 0,205	3	42.12	6	US\$ 442

* Custo horário por nó em 8 de outubro de 2020.

Custo mensal a 100% de uso por 30 dias (720 horas).

Essas opções oferecem uma capacidade de memória semelhante, mas uma capacidade e custo computacional diferentes. Para comparar os custos das suas opções específicas, consulte [Definição de preço do Amazon ElastiCache](#).

Para clusters executados no Memcached, algumas das memórias disponíveis em cada nó são usadas para sobrecarga de conexão. Para obter mais informações, consulte [Sobrecarga de conexões do Memcached](#)

O uso de vários nós exigirá a distribuição das chaves entre eles. Cada nó possui seu próprio endpoint. Para o fácil gerenciamento de endpoints, você pode usar o recurso de descoberta automática do ElastiCache, que permite que os programas clientes identifiquem automaticamente todos os nós em um cluster. Para obter mais informações, consulte [Identificar automaticamente os nós no seu cluster](#).

Em alguns casos, você pode não ter certeza de quanta capacidade precisa. Em caso afirmativo, para testes recomendamos começar com um nó cache .m5.large. Em seguida, monitore o uso da memória, a utilização da CPU e a taxa de acerto do cache com as métricas do ElastiCache publicadas no Amazon CloudWatch. Para obter mais informações sobre métricas do CloudWatch para o ElastiCache, consulte [Monitoramento de uso com métricas do CloudWatch](#). Para produção e maiores workloads, os nós R5 fornecem o melhor desempenho e valor de custo de RAM.

Se o seu cluster não tiver a taxa de acerto desejada, você poderá adicionar facilmente mais nós, aumentando assim a memória total disponível no seu cluster.

Se o seu cluster for limitado por CPU, mas tiver taxa de acerto suficiente, tente configurar um novo cluster com um tipo de nó que forneça mais poder computacional.

Criar um cluster

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

Criação de um cluster do Memcached (console)

Quando você usa o mecanismo Memcached, a Amazon ElastiCache suporta o particionamento horizontal de seus dados em vários nós. O Memcached permite a descoberta automática e, portanto, você não precisa rastrear os endpoints de cada nó. O Memcached rastreia o endpoint de cada nó, atualizando a lista de endpoints à medida que os nós são adicionados e removidos. Seu aplicativo precisa apenas do endpoint de configuração para interagir com o cluster. Para obter mais informações sobre a descoberta automática, consulte [Identificar automaticamente os nós no seu cluster](#).

Para criar um cluster do Memcached, siga as etapas em [Etapa 1: criar um cache](#)

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 [Conectar-se a nós de cache manualmente](#) e [Acesso ao seu 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](#).

Criar um cluster de cache Memcached (AWS CLI)

O código da CLI a seguir cria um cluster de cache de Memcached com 3 nós.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine memcached \  
--engine-version 1.4.24 \  
--cache-parameter-group default.memcached1.4 \  
--num-cache-nodes 3
```

Para Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine memcached ^  
--engine-version 1.4.24 ^  
--cache-parameter-group default.memcached1.4 ^  
--num-cache-nodes 3
```

Criação de um cluster (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](#).

Criação de um cluster de cache (ElastiCache API) do Memcached

O código a seguir cria um cluster Memcached com 3 nós (ElastiCache API).

As quebras de linha foram adicionadas para legibilidade.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=my-cluster  
&CacheNodeType=cache.r4.large  
&Engine=memcached  
&NumCacheNodes=3  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150508T220302Z  
&Version=2015-02-02  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Credential=<credential>  
&X-Amz-Date=20150508T220302Z  
&X-Amz-Expires=20150508T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Signature=<signature>
```

Visualização dos detalhes de um cluster

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

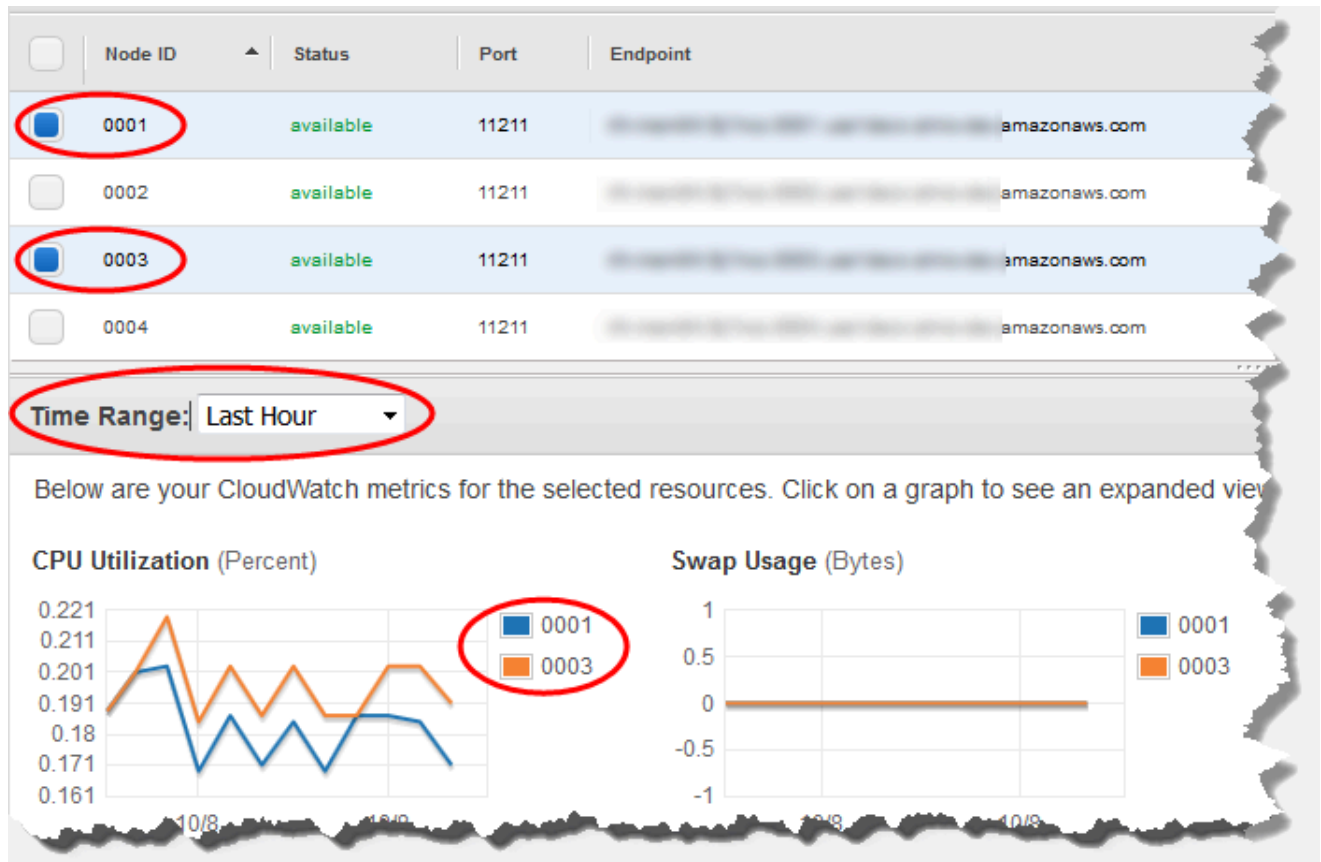
Visualização dos detalhes de um cluster (console)

Você pode visualizar os detalhes de um cluster do Memcached usando o console do ElastiCache, a AWS CLI para o ElastiCache ou a API do ElastiCache.

O procedimento a seguir detalha como visualizar os detalhes de um cluster do Memcached usando o console do ElastiCache.

Para visualizar os detalhes de um cluster Memcached

1. Faça login no 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 Memcached. Isso exibirá uma lista de todos os seus clusters que estão executando qualquer versão do Memcached.
4. Para ver os detalhes de um cluster, escolha a caixa de seleção à esquerda do nome do cluster.
5. Como visualizar informações do nó:
 - a. Escolha o nome do cluster.
 - b. Escolha a guia Nós.
 - c. Para visualizar métricas em um ou mais nós, escolha a caixa à esquerda do ID do Nó e, em seguida, escolha o intervalo de tempo para as métricas na lista Time range. Escolher vários nós gera gráficos sobrepostos.



Métricas na última hora para dois nós Memcached

Visualização dos detalhes de um cluster (AWS CLI)

Você pode visualizar os detalhes de um cluster usando o comando AWS CLI `describe-cache-clusters`. Se o parâmetro `--cache-cluster-id` for omitido, os detalhes para vários clusters, até `--max-items`, serão retornados. Se o parâmetro `--cache-cluster-id` estiver incluído, os detalhes do cluster especificado serão retornados. Você pode limitar o número de registros retornados com o parâmetro `--max-items`.

O código a seguir lista os detalhes para `my-cluster`.

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

O código a seguir lista os detalhes para até 25 clusters.

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

Example

Para Linux, macOS ou Unix:

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

Para Windows:

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

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

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Port": 11211,  
            "Address": "my-cluster.7ef-  
example.0001.usw2.cache.amazonaws.com"  
          },  
          "CacheNodeStatus": "available",  
          "ParameterGroupStatus": "in-sync",  
          "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",  
          "CustomerAvailabilityZone": "us-west-2b"  
        },  
        {  
          "CacheNodeId": "0002",  
          "Endpoint": {  
            "Port": 11211,  
            "Address": "my-cluster.7ef-  
example.0002.usw2.cache.amazonaws.com"  
          },  
          "CacheNodeStatus": "available",  
          "ParameterGroupStatus": "in-sync",  
          "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",
```

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



```
    }  
  ]  
}
```

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 determinada alteração de parâmetro é aplicada, consulte a seção Alterações entram em vigor da coluna Detalhes nas tabelas de [Parâmetros específicos do Memcached](#). Para obter informações sobre como reinicializar um cluster, consulte [Reinicialização de um cluster](#).

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:
 - Compatibilidade de versão de mecanismo
 - Grupos de segurança da VPC
 - Grupo do parâmetro
 - Janela de manutenção

- Tópico para notificação do SNS

A caixa Apply Immediately (Aplicar imediatamente) aplica-se apenas a modificações feitas na versão do mecanismo. Para aplicar alterações imediatamente, marque a caixa de seleção Apply Immediately (Aplicar imediatamente). Se essa caixa não for escolhida, as modificações feitas na versão do mecanismo serão aplicadas durante a próxima janela de manutenção. Outras modificações, como a alteração da janela de manutenção, são aplicadas imediatamente.

7. Escolha Modificar.

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. 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 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. 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](#).

Reinicialização de um cluster

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

Quando você reinicializa um cluster, este libera todos os seus dados e reinicializa seu mecanismo. Durante esse processo, você não pode acessar o cluster. Como o cluster liberou todos os seus dados, quando o cluster estiver disponível novamente, você começará com um cluster vazio.

Você pode reinicializar um cluster usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache. Se você usar o console do ElastiCache, a AWS CLI ou a API do ElastiCache, só poderá iniciar a reinicialização de um único cluster. Para reinicializar vários clusters, você deve repetir o processo ou a operação.

Usar a AWS Management Console

Você pode reinicializar um cluster usando o console do ElastiCache.

Para reinicializar 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 de interesse.
3. No painel de navegação, escolha o mecanismo em execução no cluster que deseja reinicializar.

É exibida uma lista de clusters que executam o mecanismo escolhido.

4. Selecione o cluster a ser reinicializado escolhendo o botão à esquerda do nome do cluster.

Escolha Ações e Reinicializar.

Se você escolher mais de um cluster, o botão Reinicializar não estará ativo.

Para reinicializar vários clusters, repita as etapas de 2 a 5 para cada cluster que você deseja reinicializar. Você não precisa esperar que um cluster termine a reinicialização para reinicializar outro.

Para reinicializar um nó específico, selecione o nó e escolha Reboot (Reinicializar).

Usar a AWS CLI

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

Para reinicializar nós específicos no cluster, use o `--cache-node-ids-to-reboot` para listar os clusters específicos para reinicializar. O comando a seguir reinicializa os nós 0001, 0002 e 0004 de `my-cluster`.

Para Linux, macOS ou Unix:

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --cache-node-ids-to-reboot 0001 0002 0004
```

Para Windows:

```
aws elasticache reboot-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --cache-node-ids-to-reboot 0001 0002 0004
```

Para reinicializar todos os nós no cluster, use o parâmetro `--cache-node-ids-to-reboot` e liste todos os ID de nós do cluster. Para obter mais informações, consulte [reboot-cache-cluster](#).

Uso da API do ElastiCache

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

Para reinicializar nós específicos no cluster, use o `CacheNodeIdsToReboot` para listar os clusters específicos para reinicializar. O comando a seguir reinicializa os nós 0001, 0002 e 0004 de `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RebootCacheCluster  
&CacheClusterId=my-cluster  
&CacheNodeIdsToReboot.member.1=0001  
&CacheNodeIdsToReboot.member.2=0002  
&CacheNodeIdsToReboot.member.3=0004  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Para reinicializar todos os nós no cluster, use o parâmetro `CacheNodeIdsToReboot` e liste todos os ID de nós do cluster. Para obter mais informações, consulte [RebootCacheCluster](#).

Adição de nós a um cluster

Adicionar nós a um cluster do Memcached aumenta o número de partições dele. Quando você altera o número de partições em um cluster, alguns dos principais espaços precisam ser remapeados para que sejam mapeados para o nó direito. Remapear espaços de chave aumenta temporariamente o número de falhas de cache no cluster. Para obter mais informações, consulte [Configuração do seu cliente do ElastiCache para balanceamento de carga eficiente](#).

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

Tópicos

- [Para adicionar nós a um cluster \(console\)](#)

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.
4. Escolha Adicionar nó.
5. Preencha as informações solicitadas na caixa de diálogo Add Node (Adicionar nó).
6. Escolha o botão Apply Immediately - Yes (Aplicar imediatamente - sim) para aplicar essa alteração imediatamente ou escolha No (Não) para adicionar este nó durante a próxima janela de manutenção do cluster.

Impacto de novas solicitações de adição e remoção em solicitações pendentes

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 1	Delete	Delete	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para excluir os nós 0002 e 0004 for emitida, somente os nós 0002 e 0004 serão excluídos. Os nós 0001, 0003 e 0007 não serão excluídos.</p>
Cenário 2	Delete	Criar	<p>A nova solicitação de criação, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para criar um nó for emitida, um novo nó será criado e os nós 0001, 0003 e 0007 não serão excluídos.</p>
Cenário 3	Criar	Delete	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de criação pendente.</p> <p>Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para excluir o nó 0003, nenhum novo nó será criado e o nó 0003 será excluído.</p>

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 4	Criar	Criar	<p>A nova solicitação de criação é adicionada à solicitação de criação pendente.</p> <p>Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para criar três nós, as novas solicitações serão adicionadas à solicitação pendente e cinco nós serão criados.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <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

Para adicionar nós a um cluster usando a AWS CLI, use a AWS CLI operação `modify-cache-cluster` com os seguintes parâmetros:

- `--cache-cluster-id`: o ID do cluster de cache ao qual você deseja adicionar os nós.
- `--num-cache-nodes`: o parâmetro `--num-cache-nodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada. Para adicionar nós a esse cluster, `--num-cache-nodes` deve ser maior que o número atual de nós neste cluster. Se esse valor for inferior ao número atual de nós, o ElastiCache esperará o parâmetro `cache-node-ids-to-remove` e uma lista de nós para remover do cluster. Para obter mais informações, consulte [Usar a AWS CLI](#).
- `--apply-immediately` ou `--no-apply-immediately`, que especifica se você deve adicionar esses nós imediatamente ou na próxima janela de manutenção.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

Para Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

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

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",
```

```
    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
      "Port": 11211,
      "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "NumCacheNodes": 2,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
      "NumCacheNodes": 5
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
  }
}
```

Para obter mais informações, consulte o tópico sobre a AWS CLI [modify-cache-cluster](#).

Uso da API do ElastiCache

Para adicionar nós a um cluster (API do ElastiCache)

- Chame a operação de API `ModifyCacheCluster` com os seguintes parâmetros:
 - `CacheClusterId`: o ID do cluster ao qual você deseja adicionar os nós.
 - `NumCacheNodes`: o parâmetro `NumCachNodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada. Para adicionar nós a esse cluster, `NumCacheNodes` deve ser maior que o número atual de nós neste cluster. Se esse valor for inferior ao número atual de nós, o ElastiCache esperará o parâmetro

CacheNodeIdsToRemove com uma lista de nós para remover do cluster (consulte [Uso da API do ElastiCache](#)).

- `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=ModifyCacheCluster  
  &ApplyImmediately=true  
  &NumCacheNodes=5  
&CacheClusterId=my-cluster  
&Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico da API do ElastiCache [ModifyCacheCluster](#).

Remoção de nós de um cluster

Cada vez que você altera o número de nós em um cluster Memcached, você deve remapear pelo menos alguns dos seus espaços de chaves para que ele seja mapeado para o nó correto. Para obter informações mais detalhadas sobre o balanceamento de carga de um cluster Memcached, consulte [Configuração do seu cliente do ElastiCache para balanceamento de carga eficiente](#).

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.

Impacto de novas solicitações de adição e remoção em solicitações pendentes

Cenários	Operação pendente	Nova solicitação	Resultados
Cenário 1	Delete	Delete	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para excluir os nós 0002 e 0004 for emitida, somente os nós 0002 e 0004 serão excluídos. Os nós 0001, 0003 e 0007 não serão excluídos.</p>
Cenário 2	Delete	Criar	<p>A nova solicitação de criação, pendente ou imediata, substitui a solicitação de exclusão pendente.</p> <p>Por exemplo, se os nós 0001, 0003 e 0007 estiverem pendentes de exclusão e uma nova solicitação para criar um nó for emitida, um novo nó será criado e os nós 0001, 0003 e 0007 não serão excluídos.</p>
Cenário 3	Criar	Delete	<p>A nova solicitação de exclusão, pendente ou imediata, substitui a solicitação de criação pendente.</p> <p>Por exemplo, se houver uma solicitação pendente para criar dois nós e uma nova solicitação for emitida para excluir o nó 0003, nenhum novo nó será criado e o nó 0003 será excluído.</p>
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
			<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>

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 `modify-cache-cluster` 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 `modify-cache-cluster` com os seguintes parâmetros:

- `--cache-cluster-id`: o ID do cluster de cache do qual você deseja remover nós.
- `--num-cache-nodes`: o parâmetro `--num-cache-nodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada.
- `--cache-node-ids-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.

O exemplo a seguir remove imediatamente o nó 0001 do cluster my-cluster.

Para Linux, macOS ou Unix:

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

Para Windows:

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

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

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",  
    "PreferredAvailabilityZone": "us-east-2b",  
    "ConfigurationEndpoint": {  
      "Port": 11211,  
      "Address": "rlh-mem000.7ef-example.cfg.usw2.cache.amazonaws.com"  
    },  
    "CacheSecurityGroups": [],  
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z", 9dcv5r  
    "AutoMinorVersionUpgrade": true,  
    "CacheClusterStatus": "modifying",
```

```

    "NumCacheNodes": 3,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
      "NumCacheNodes": 2,
      "CacheNodeIdsToRemove": [
        "0001"
      ]
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
  }
}

```

Para obter mais informações, consulte os AWS CLI tópicos [describe-cache-cluster](#) e [modify-cache-cluster](#).

Uso da API do ElastiCache

Para remover nós usando a API do ElastiCache, chame a operação da API `ModifyCacheCluster` com o ID do cluster de cache e uma lista de nós para remoção, conforme mostrado:

- `CacheClusterId`: o ID do cluster de cache do qual você deseja remover nós.
- `NumCacheNodes`: o parâmetro `NumCacheNodes` especifica o número de nós que você deseja neste cluster após a modificação ser aplicada.
- `CacheNodeIdsToRemove.member.n`: A lista de IDs de nó a serem removidos do cluster.
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately`: Especifica se você deve remover esses nós imediatamente ou na próxima janela de manutenção.

- **Region:** especifica a 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=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&ApplyImmediately=true  
&CacheNodeIdsToRemove.member.1=0004  
&CacheNodeIdsToRemove.member.2=0005  
&NumCacheNodes=3  
&Region us-east-2  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Para obter mais informações, consulte o tópico da API do ElastiCache [ModifyCacheCluster](#).

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.

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 Ações, escolha Excluir.
5. Na tela de confirmação Excluir cluster, escolha Excluir para excluir o cluster ou Cancelar para mantê-lo.

Se você escolheu Excluir, o status do cluster muda para excluindo.

Assim que o cluster não estiver mais relacionado na lista de clusters, você para de ser cobrado por ele.

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

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](#)

Concessão de acesso ao cluster

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 Memcached é **11211**.
 - 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 Memcached com:
 - Endereço IP: 10.0.1.230
 - Porta Memcached padrão: 11211
 - Grupo de segurança: *10\0\0\55*
- 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 (11211 - 11213).
 - 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	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

- Uma regra de saída para permitir conexões TCP para a porta de cache (11211).

Grupo de segurança da instância NAT: regra de saída

Tipo	Protocolo	Intervalo de portas	Destino
Regra personalizada de TCP	TCP	11211	sg-ce56b7a9 (grupo de segurança 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 (11211).

Grupo de segurança da instância do cluster: regra de entrada

Tipo	Protocolo	Intervalo de portas	Origem
Regra personalizada de TCP	TCP	11211	sg-bd56b7da (grupo de segurança NAT)

3. Valide as regras.

- Confirme se o cliente confiável é capaz de fazer SSH para a instância NAT.
- Confirme se o cliente confiável pode se conectar ao cluster a partir da instância NAT.

4. Adicione uma regra iptables à instância NAT.

Uma regra iptables deve ser adicionada à tabela NAT para cada nó no cluster para encaminhar a porta de cache da instância NAT ao nó do cluster. Um exemplo pode ser o seguinte:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
```

O número da porta deve ser exclusivo para cada nó no cluster. Por exemplo, se estiver trabalhando com um cluster Memcached de três nós usando as portas 11211 - 11213, as regras seriam as seguintes:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to 10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to 10.0.1.232:11211
```

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:

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

Um cliente telnet também pode ser usado para verificar a conexão. Por exemplo:

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

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.

Quais endpoints usar

Para o cache sem servidor do ElastiCache com o Memcached, basta adquirir o DNS do endpoint do cluster e a porta do console.

Na AWS CLI, use o comando `describe-serverless-caches` para adquirir as informações do endpoint.

Linux

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

Windows

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

O resultado da operação acima deve ser semelhante a este (formato JSON).

```
{
  "ServerlessCaches": [
    {
      "ServerlessCacheName": "serverless-memcached",
      "Description": "test",
      "CreateTime": 1697659642.136,
      "Status": "available",
      "Engine": "memcached",
      "MajorEngineVersion": "1.6",
      "FullEngineVersion": "21",
      "SecurityGroupIds": [
        "sg-083eda453e1e51310"
      ],
      "Endpoint": {
        "Address": "serverless-memcached-01.amazonaws.com",
        "Port": 11211
      },
      "ARN": "<the ARN>",
      "SubnetIds": [
        "subnet-0cf759df15bd4dc65",
        "subnet-09e1307e8f1560d17"
      ],
      "SnapshotRetentionLimit": 0,
      "DailySnapshotTime": "03:00"
    }
  ]
}
```

Para um cluster do Memcached com base na instância, se usar a descoberta automática, você poderá usar o endpoint de configuração do cluster para configurar o cliente do Memcached. Isso significa que você deve usar um cliente que ofereça suporte para Descoberta automática.

Se você não usar a Descoberta automática, deverá configurar seu cliente para usar os endpoints de nó individuais para leituras e gravações. Você também deve acompanhá-los à medida que adiciona e remove nós.

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 (console)

Todos os endpoints Memcached são endpoints de leitura/gravação. Para se conectar a nós em um cluster Memcached, seu aplicativo pode usar os endpoints para cada nó ou o endpoint de configuração do cluster, juntamente com a Descoberta automática. Para usar a Descoberta automática, você deve usar um cliente que ofereça suporte para Descoberta automática.

Ao usar a Descoberta automática, seu aplicativo cliente se conecta ao seu cluster Memcached usando o endpoint de configuração. À medida que você escalar seu cluster adicionando ou removendo nós, seu aplicativo "conhecerá" automaticamente todos os nós do cluster e poderá se conectar a qualquer um deles. Sem a Descoberta automática, seu aplicativo teria que fazer isso, ou você precisaria atualizar manualmente os endpoints no seu aplicativo sempre que adicionasse ou removesse um nó.

Para copiar um endpoint, escolha o ícone de cópia diretamente na frente do endereço do endpoint. Para obter informações sobre como usar o endpoint para se conectar a um nó, consulte [Conexão a nós](#).

Endpoints de configuração e de nó são muitos semelhantes. As diferenças estão realçadas com negrito a seguir.

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port # configuration endpoint  
contains "cfg"  
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

Important

Se você optar por criar um CNAME para o seu endpoint de configuração Memcached, para que o seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir `.cfg.` nesse CNAME.

Localizar endpoints (AWS CLI)

Você pode usar a AWS CLI do Amazon ElastiCache para descobrir os endpoints para nós e clusters.

Tópicos

- [Encontrar endpoints para nós e clusters \(AWS CLI\)](#)

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 Memcached, o comando retorna o endpoint de configuração. Se você incluir o parâmetro opcional `--show-cache-node-info`, o comando também retornará os endpoints dos nós individuais no cluster.

Example

O comando a seguir recupera os endpoints de configuração (`ConfigurationEndpoint`) e os endpoints de nó individuais (`Endpoint`) para o cluster Memcached `mycluster`.

Para Linux, macOS ou Unix:

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

Para Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

O resultado da operação acima deve ser semelhante a este (formato JSON).

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {  
          "CacheNodeId": "0001",  
          "Endpoint": {
```

```
    "Port": 11211,
    "Address": "mycluster.amazonaws.com"
  },
  "CacheNodeStatus": "available",
  "ParameterGroupStatus": "in-sync",
  "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
  "CustomerAvailabilityZone": "us-west-2b"
},
{
  "CacheNodeId": "0002",
  "Endpoint": {
    "Port": 11211,
    "Address": "mycluster.amazonaws.com"
  },
  "CacheNodeStatus": "available",
  "ParameterGroupStatus": "in-sync",
  "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
  "CustomerAvailabilityZone": "us-west-2b"
},
{
  "CacheNodeId": "0003",
  "Endpoint": {
    "Port": 11211,
    "Address": "mycluster.amazonaws.com"
  },
  "CacheNodeStatus": "available",
  "ParameterGroupStatus": "in-sync",
  "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
  "CustomerAvailabilityZone": "us-west-2b"
}
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "us-west-2b",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
```

```
    "AutoMinorVersionUpgrade": true,  
    "CacheClusterStatus": "available",  
    "NumCacheNodes": 3,  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheSubnetGroupName": "default",  
    "EngineVersion": "1.4.24",  
    "PendingModifiedValues": {},  
    "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",  
    "CacheNodeType": "cache.m4.large",  
  
  }  
]  
}
```

Important

Se você optar por criar um CNAME para o seu endpoint de configuração Memcached, para que o seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir `.cfg.` nesse CNAME. Por exemplo, `mycluster.cfg.local` no seu arquivo `php.ini` para o parâmetro `session.save_path`.

Para obter mais informações, consulte o tópico [describe-cache-clusters](#).

Localização de endpoints (API do ElastiCache)

Você pode usar a API do Amazon ElastiCache para descobrir os endpoints para nós e clusters.

Tópicos

- [Localização de endpoints para nós e clusters \(API do ElastiCache\)](#)

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 Memcached, o comando retorna o endpoint de configuração. Se você incluir o parâmetro opcional `ShowCacheNodeInfo`, a ação também retornará os endpoints dos nós individuais no cluster.

Example

O comando a seguir recupera os endpoints de configuração (`ConfigurationEndpoint`) e os endpoints de nó individuais (`Endpoint`) para o cluster Memcached `mycluster`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=mycluster  
  &ShowCacheNodeInfo=true  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

Important

Se você optar por criar um CNAME para o seu endpoint de configuração Memcached, para que o seu cliente de descoberta automática reconheça o CNAME como um endpoint de configuração, você deverá incluir `.cfg` nesse CNAME. Por exemplo, `mycluster.cfg.local` no seu arquivo `php.ini` para o parâmetro `session.save_path`.

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 foi criado ou modificado pela última vez. Cada nó possui seu próprio nome DNS (Serviço de Nomes de Domínio) e porta. Há suporte para vários tipos de nós do ElastiCache, cada um com quantidades diversificadas de memória associada e a potência computacional.

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

Tópicos

- [Visualizando o status ElastiCache do nó](#)
- [Conexão a nós](#)
- [Tipos de nó compatíveis](#)
- [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 Memcached](#)
- [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 nóculo está saudável e disponível.
creating	Não faturado	O ElastiCache nó está sendo criado. O nó está inacessível enquanto é criado.
deleting	Não faturado	O ElastiCache nó está sendo excluído.
modifying	Faturado	O ElastiCache nó está sendo modificado devido a uma solicitação do cliente para modificar o nó.
updating	Faturado	<p>Um estado de atualização indica que uma ou mais das seguintes afirmações são verdadeiras para o ElastiCache nó da Amazon:</p> <ul style="list-style-type: none">• O ElastiCache nó está sendo corrigido como parte da atualização do serviço. Para obter mais informações sobre as atualizações do serviço, consulte a página de ajuda do Amazon ElastiCache Managed Maintenance and Service Updates.• Os grupos de segurança da VPC estão sendo atualizados para o ElastiCache cluster.

Tipo	Faturado	Descrição
		<ul style="list-style-type: none">• 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
rebooting cache cluster nodes	Faturado	O ElastiCache nó está sendo reinicializado devido a uma solicitação do cliente ou a um ElastiCache processo da Amazon que exige a reinicialização do nó.

Tipo	Faturado	Descrição
incompatible_parameters	Não faturado	<p>A Amazon não ElastiCache pode iniciar o nó porque os parâmetros especificados no grupo de parâmetros do nó não são compatíveis com o nó. Reverta as alterações de parâmetro ou as torne compatíveis com o nó para retomar o acesso ao nó. Para obter mais informações sobre os parâmetros incompatíveis, consulte a lista de eventos do ElastiCache nó.</p>
incompatible_network	Não faturado	<p>Um estado de rede incompatível indica que uma ou mais das seguintes afirmações são verdadeiras para o nó da Amazon: ElastiCache</p> <ul style="list-style-type: none">• 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.

Conexão a nós

Antes de tentar se conectar ao seu cluster Memcached, você deve ter os endpoints para os nós. Para localizar os endpoints, consulte o seguinte:

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

No exemplo a seguir, você usa o utilitário telnet para se conectar a um nó que está executando o Memcached.

Note

Para obter mais informações sobre o Memcached e os comandos Memcached disponíveis, consulte o site do [Memcached](#).

Para se conectar a um nó usando o telnet

1. Conecte-se à instância do Amazon EC2 com o utilitário de conexão de sua escolha.

Note

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

2. Baixe e instale o utilitário telnet na sua instância do Amazon EC2. No prompt de comando da sua instância do Amazon EC2, digite o seguinte comando e digite y.

```
sudo yum install telnet
```

É exibida uma saída semelhante à seguinte.

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

```
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB      00:00

...(output omitted)...

Complete!
```

3. No prompt de comando da sua instância do Amazon EC2, digite o seguinte comando, substituindo o endpoint do seu nó por aquele mostrado neste exemplo.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

É exibida uma saída semelhante à seguinte.

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

4. Teste a conexão executando comandos Memcached.

Agora você está conectado a um nó e pode executar comandos Memcached. Veja um exemplo a seguir.

```
set a 0 0 5      // Set key "a" with no expiration and 5 byte value
hello           // Set value as "hello"
STORED
get a           // Get value for key "a"
VALUE a 0 5
hello
END
get b           // Get value for key "b" results in miss
END
>
```


Tipos de nó compatíveis

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

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](#).

- Propósito geral:
- Geração atual:

Tipos de nó M6g (disponíveis somente para o mecanismo do Memcached versão 1.5.16 em diante).

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

Note

Para obter a disponibilidade das regiões, consulte [Tipos de nós compatíveis por região da AWS](#).

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

Tipos de nó T4g (disponíveis somente para o mecanismo do Memcached versão 1.5.16 em diante).

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

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

Tipos de nó T2: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`

- Geração anterior: (não recomendada. Os clusters existentes ainda são compatíveis, mas não há suporte para a criação de novos clusters para esses tipos.)

Tipos de nó T1: `cache.t1.micro`

Tipos de nó M1: `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`,
`cache.m1.xlarge`

Tipos de nó M3: `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`,
`cache.m3.2xlarge`


- Otimizadas para computação:
 - Geração anterior: (não recomendada)

Tipos de nó C1: `cache.c1.xlarge`

- Otimizado para memória:
 - Geração atual:

(Os tipos de nó R6g estão disponíveis somente para o mecanismo do Memcached versão 1.5.16 em diante).

Tipos de nó R6g: `cache.r6g.large`, `cache.r6g.xlarge`, `cache.r6g.2xlarge`,
`cache.r6g.4xlarge`, `cache.r6g.8xlarge`, `cache.r6g.12xlarge`,
`cache.r6g.16xlarge`

 Note

Para obter a disponibilidade das regiões, consulte [Tipos de nós compatíveis por região da AWS](#).

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

- Geração anterior: (não recomendada)

Tipos de nó M2: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Tipos de nó R3: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`,
`cache.r3.4xlarge`, `cache.r3.8xlarge`

- Otimizadas para rede:
- Geração atual:

(Os tipos de nó C7gn estão disponíveis somente para o mecanismo do Memcached versão 1.6.6 em diante).

Tipos de nó C7gn: `cache.c7gn.large`, `cache.c7gn.xlarge`, `cache.c7gn.2xlarge`,
`cache.c7gn.4xlarge`, `cache.c7gn.8xlarge`, `cache.c7gn.12xlarge`,
`cache.c7gn.16xlarge`

Geração atual

As tabelas a seguir mostram a largura de banda de linha de base e de expansão que utilizam o mecanismo de crédito de E/S de rede para ultrapassar a largura de banda de linha de base.

Note

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

Geral

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
<code>cache.m7g.large</code>		0.937	12,5
<code>cache.m7g.xlarge</code>		1.876	12,5
<code>cache.m7g.2xlarge</code>		3,75	15

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m7g.4xlarge		7,5	15
cache.m7g.8xlarge		15	N/D
cache.m7g.12xlarge		22.5	N/D
cache.m7g.16xlarge		30	N/D
cache.m6g.large	1.5.16	0.75	10.0
cache.m6g.xlarge	1.5.16	1.25	10.0
cache.m6g.2xlarge	1.5.16	2,5	10.0
cache.m6g.4xlarge	1.5.16	5,0	10.0
cache.m6g.8xlarge	1.5.16	12	N/D
cache.m6g.12xlarge	1.5.16	20	N/D
cache.m6g.16xlarge	1.5.16	25	N/D
cache.m5.large	1.5.16	0.75	10.0
cache.m5.xlarge	1.5.16	1.25	10.0
cache.m5.2xlarge	1.5.16	2,5	10.0
cache.m5.4xlarge	1.5.16	5,0	10.0
cache.m5.12xlarge	1.5.16	N/D	N/D
cache.m5.24xlarge	1.5.16	N/D	N/D
cache.m4.large	1.5.16	0.45	1.2
cache.m4.xlarge	1.5.16	0.75	2.8

Tipo de instância	Versão mínima do Memcached compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.m4.2xlarge	1.5.16	1,0	10,0
cache.m4.4xlarge	1.5.16	2,0	10,0
cache.m4.10xlarge	1.5.16	5,0	10,0
cache.t4g.micro	1.5.16	0.064	5,0
cache.t4g.small	1.5.16	0.128	5,0
cache.t4g.medium	1.5.16	0.256	5,0
cache.t3.micro	1.5.16	0.064	5,0
cache.t3.small	1.5.16	0.128	5,0
cache.t3.medium	1.5.16	0.256	5,0
cache.t2.micro	1.5.16	0.064	1.024
cache.t2.small	1.5.16	0.128	1.024
cache.t2.medium	1.5.16	0.256	1.024

Otimizado para memória

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r7g.large		0.937	12,5
cache.r7g.xlarge		1.876	12,5
cache.r7g.2xlarge		3,75	15
cache.r7g.4xlarge		7,5	15

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r7g.8xlarge		15	N/D
cache.r7g.12xlarge		22.5	N/D
cache.r7g.16xlarge		30	N/D
cache.r6g.large	1.5.16	0.75	10.0
cache.r6g.xlarge	1.5.16	1.25	10.0
cache.r6g.2xlarge	1.5.16	2,5	10.0
cache.r6g.4xlarge	1.5.16	5,0	10.0
cache.r6g.8xlarge	1.5.16	12	N/D
cache.r6g.12xlarge	1.5.16	20	N/D
cache.r6g.16xlarge	1.5.16	25	N/D
cache.r5.large	1.5.16	0.75	10.0
cache.r5.xlarge	1.5.16	1.25	10.0
cache.r5.2xlarge	1.5.16	2,5	10.0
cache.r5.4xlarge	1.5.16	5,0	10.0
cache.r5.12xlarge	1.5.16	20	N/D
cache.r5.24xlarge	1.5.16	25	N/D
cache.r4.large	1.5.16	0.75	10.0
cache.r4.xlarge	1.5.16	1.25	10.0
cache.r4.2xlarge	1.5.16	2,5	10.0
cache.r4.4xlarge	1.5.16	5,0	10.0

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.r4.8xlarge	1.5.16	12	N/D
cache.r4.16xlarge	1.5.16	25	N/D

Otimizadas para rede

Tipo de instância	Versão mínima compatível	Largura de banda da linha de base (Gbps)	Expansão da largura de banda (Gbps)
cache.c7gn.large	1.6.6	6.25	30
cache.c7gn.xlarge	1.6.6	12,5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/D
cache.c7gn.8xlarge	1.6.6	100	N/D
cache.c7gn.12xlarge	1.6.6	150	N/D
cache.c7gn.16xlarge	1.6.6	200	N/D

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

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
24	576	2	20%	3,09	Até 5 gigabits
6	144	1	10%	0,5	Baixo a moderado
12	288	1	20%	1.55	Baixo a moderado
24	576	2	20%	3.22	Baixo a moderado

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

** O desempenho de linha de base na tabela é por vCPU. Alguns tamanhos de nós que têm mais de uma vCPU. Para 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.

Informações relacionadas

- [Recursos e detalhes do produto Amazon ElastiCache](#)
- [Parâmetros específicos do tipo de nó Memcached](#)

Substituição de nós

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

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

Note

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

A lista a seguir identifica as ações que você pode realizar quando o ElastiCache programa um de seus nós do Memcached para substituição.

- Não fazer nada: se você não fizer nada, o ElastiCache substituirá o nó conforme programado. Quando o ElastiCache substitui automaticamente o nó por um novo nó, o novo nó fica inicialmente vazio.
- Mudar sua janela de manutenção para eventos de manutenção programada, você recebe um email ou uma notificação do evento do ElastiCache. Nesse caso, se você mudar sua janela de manutenção antes da hora de substituição programada, o nó será substituído no novo horário. Para obter mais informações, consulte [Modificando um cluster ElastiCache](#).

Note

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

Se a notificação não inclui uma janela de manutenção, não é possível alterar a janela de substituição.

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

- Você altera sua janela de manutenção para sexta-feira, 16h (após a data e hora atual e antes da próxima janela de manutenção programada). O nó é substituído na sexta-feira, 10 de novembro, às 16h.
- Você altera sua janela de manutenção para sábado, 16h (após a data e hora atual e a próxima janela de manutenção programada). O nó é substituído no sábado, 11 de novembro, às 16h.
- Você altera sua janela de manutenção para quarta-feira, 16h (dia da semana anterior à data e hora atual). O nó é substituído na próxima quarta-feira, 15 de novembro, às 16h.

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

- Substituir manualmente o nó se você precisar substituir o nó antes da próxima janela de manutenção, substitua o nó manualmente.

Se você substituir o nó manualmente, as chaves serão redistribuídas. Essa redistribuição causa perdas no cache.

Para substituir manualmente um nó Memcached

1. Exclua o nó agendado para substituição. Para obter instruções, consulte [Remoção de nós de um cluster](#).
2. Adicione um novo nó ao cluster. Para obter instruções, consulte [Adição de nós a um cluster](#).
3. Se você não estiver usando a descoberta automática neste cluster, consulte a aplicação e substitua todas as instâncias do endpoint do nó anterior pelo endpoint do novo nó.

Nós reservados 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 Memcached.
5. Para determinar as ofertas disponíveis, faça seleções nas seguintes opções:
 - Node Type
 - Prazo
 - Offering Type

Após essas seleções, o custo por nó e o custo total de suas seleções serão exibidos em Reservation details (Detalhes da reserva).

6. Escolha Cancel para evitar a compra desses nós e gerar cobranças.

Obtenção de informações sobre ofertas de nós reservados (AWS CLI)

Para obter 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": "memcached",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "memcached",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xx.12xlarge",
  "Duration": 31536000,
```

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

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

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

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

Example

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

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

Compra de um nó reservado

Os exemplos a seguir mostram como comprar uma oferta de nó reservado usando o 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 Memcached.
5. Para determinar as ofertas disponíveis, faça seleções nas seguintes opções:
 - Node Type
 - Prazo
 - Offering Type
 - Um Reserved node ID (ID de nó reservado) opcional

Após essas seleções, o custo por nó e o custo total de suas seleções serão exibidos em Reservation details (Detalhes da reserva).

6. Escolha Purchase (Comprar).

Compra de um nó reservado (AWS CLI)

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

Digite o seguinte comando em um prompt de comando:

Para Linux, macOS ou Unix:

```
aws elasticache purchase-reserved-cache-nodes-offering \  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \  
  --reserved-cache-node-id myreservationID
```

Para Windows:

```
aws elasticache purchase-reserved-cache-nodes-offering ^ \  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^ \  
  --reserved-cache-node-id myreservationID
```

Esse comando retorna uma saída semelhante à seguinte:

RESERVATION	ReservationId	Class	Start Time	Duration	
Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	myreservationid	cache.xx.small	2013-12-19T00:30:23.247Z	1y	
XXX.XX USD	X.XXX USD	1	payment-pending	memcached	Medium Utilization

Para obter mais informações, consulte [purchase-reserved-cache-nodes-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",
```

```
"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 Memcached

Para migrar o ElastiCache for Memcached para um tipo de nó diferente, você deve criar um novo cluster, que sempre começa vazio, que sua aplicação possa preencher.

Para migrar o tipo de nó do cluster do ElastiCache for Memcached usando o console do ElastiCache:

- Crie um novo cluster com o novo tipo de nó. Para obter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).
- No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para obter mais informações, consulte [Localização de endpoints de um cluster \(console\)](#)
- Exclua o cluster antigo. Para obter mais informações, consulte [Excluir um cluster](#).

Trabalhando com ElastiCache

Nesta seção, você pode encontrar detalhes sobre como gerenciar os vários componentes da sua ElastiCache implementação.

Tópicos

- [Versões de mecanismos e atualização](#)
- [Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache](#)
- [Gerenciamento do cluster autoprojeto](#)
- [Dimensionamento ElastiCache para Memcached](#)
- [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](#)

Versões de mecanismos e atualização

Esta seção aborda as versões compatíveis do mecanismo Memcached e como atualizar.

Tópicos

- [Versões do ElastiCache para Memcached com suporte](#)
- [Versões de mecanismos e atualização](#)
- [Como atualizar as versões dos mecanismos](#)

Versões do ElastiCache para Memcached com suporte

O ElastiCache oferece suporte às seguintes versões do Memcached e às atualizações para as versões mais recentes. Ao atualizar para uma versão mais recente, fique atento às condições que, caso não sejam atendidas, farão com que a atualização falhe.

Versões do ElastiCache para Memcached

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

Memcached versão 1.6.22

O ElastiCache para Memcached adiciona suporte para o Memcached versão 1.6.22. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.6.18](#).

Para obter mais informações, consulte [ReleaseNotes1622](#) do Memcached no GitHub.

Memcached versão 1.6.17

O ElastiCache para Memcached adiciona suporte para o Memcached versão 1.6.17. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.6.17](#).

Para obter mais informações, consulte [ReleaseNotes1617](#) do Memcached no GitHub.

Memcached versão 1.6.12

O ElastiCache para Memcached adiciona suporte para Memcached versão 1.6.12 e criptografia em trânsito. Ele inclui correções de erros e atualizações cumulativas do [Memcached 1.6.6](#).

Para obter mais informações, consulte [ReleaseNotes1612](#) do Memcached no GitHub.

Memcached versão 1.6.6

O ElastiCache para Memcached adiciona suporte para Memcached versão 1.6.6. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas do [Memcached 1.5.16](#). O ElastiCache para Memcached não inclui suporte para [Extstore](#).

Para obter mais informações, consulte [ReleaseNotes166](#) no Memcached no GitHub.

Memcached versão 1.5.16

O ElastiCache para Memcached adiciona suporte para Memcached versão 1.5.16. Não inclui novos recursos, mas inclui correções de erros e atualizações cumulativas de [Memcached 1.5.14](#) e [Memcached 1.5.15](#).

Para obter mais informações, consulte [Notas de release do Memcached 1.5.16](#) no Memcached no GitHub.

Memcached versão 1.5.10

O ElastiCache para Memcached versão 1.5.10 oferece suporte para os seguintes recursos do Memcached:

- Rebalanceamento de slab automatizado.
- Pesquisas de tabela de hash mais rápidas com o algoritmo murmur3.
- Algoritmo LRU segmentado.
- Crawler de LRU para memória de recuperação do plano de fundo.
- `--enable-seccomp`: uma opção de tempo de compilação.

Ele também apresenta os parâmetros `no_modern` e `inline_ascii_resp`. Para obter mais informações, consulte [Alterações de parâmetros do Memcached 1.5.10](#).

As melhorias no Memcached adicionadas desde o ElastiCache para o Memcached versão 1.4.34 incluem os seguintes:

- Correções cumulativas, como os multiget ASCII, CVE-2017-9951 e limites de crawls para o `metadumper`.
- Melhor gerenciamento de conexão ao fechar as conexões no limite da conexão.

- Gerenciamento do tamanho do item aprimorado para tamanho do item acima de 1MB.
- Melhor desempenho e melhorias de sobrecarga da memória reduzindo os requisitos de memória por item por alguns bytes.

Para obter mais informações, consulte [Notas de release do Memcached 1.5.10](#) no Memcached no GitHub.

Memcached versão 1.4.34

O ElastiCache para Memcached versão 1.4.34 não adiciona novos recursos à versão 1.4.33. A versão 1.4.34 é uma versão da correção de bugs maior que a versão usual.

Para obter mais informações, consulte [Notas de release do Memcached 1.4.34](#) no Memcached no GitHub.

Memcached versão 1.4.33

As melhorias no Memcached adicionadas desde a versão 1.4.24 incluem:

- Capacidade de despejar todos os metadados de uma classe de slab específica, uma lista de classes de slab ou todas as classes de slab. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.31](#).
- Melhor suporte para itens grandes acima do padrão de 1 megabyte. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.29](#).
- Capacidade de especificar por quanto tempo um cliente pode estar ocioso antes de ser solicitado a fechar.

Capacidade de aumentar dinamicamente a quantidade de memória disponível para o Memcached sem ter que reiniciar o cluster. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.27](#).

- Agora há suporte para o registro em log de fetchers, mutations e evictions. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.26](#).
- A memória liberada pode ser recuperada para um grupo global e reatribuída a novas classes de slab. Para obter mais informações, consulte o documento [Notas de release do Memcached 1.4.25](#).
- Diversas correções de bugs.
- Alguns novos comandos e parâmetros. Para ver uma lista, consulte [Parâmetros adicionados do Memcached 1.4.33](#).

Memcached versão 1.4.24

As melhorias no Memcached adicionadas desde a versão 1.4.14 incluem:

- Gerenciamento de LRUs (menos utilizados recentemente) usando um processo em segundo plano.
- Adição da opção de usar o jenkins ou o murmur3 como algoritmo de hash.
- Alguns novos comandos e parâmetros. Para ver uma lista, consulte [Parâmetros adicionados do Memcached 1.4.24](#).
- Diversas correções de bugs.

Memcached versão 1.4.14

As melhorias no Memcached adicionadas desde a versão 1.4.5 incluem:

- Maior capacidade de rebalanceamento de slabs.
- Melhoria de desempenho e escalabilidade.
- Introdução do comando touch para atualizar o tempo de expiração de um item existente sem buscá-lo.
- Descoberta automática: a capacidade de programas clientes determinarem automaticamente todos os nós do cache em um cluster e iniciarem e manterem conexões com todos esses nós.

Memcached versão 1.4.5

O Memcached versão 1.4.5 era o mecanismo e versão inicial com suporte pelo Amazon ElastiCache para Memcached.

Versões de mecanismos e atualização

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.

Para atualizar para uma versão do Memcached mais recente, modifique seu cluster de cache especificando a nova versão do mecanismo que você deseja usar. Atualizar para uma versão do Memcached mais recente é um processo destrutivo. Você perde seus dados e começa com um cache frio. Para obter mais informações, consulte [Modificação de clusters](#).

Você deve estar ciente dos seguintes requisitos ao atualizar de uma versão mais antiga do Memcached para o Memcached versão 1.4.33 ou posterior. `CreateCacheCluster` e `ModifyCacheCluster` falham nas seguintes condições:

- Se `slab_chunk_max > max_item_size`.
- Se `max_item_size` modulo `slab_chunk_max` $\neq 0$.
- Se `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

O valor `(max_cache_memory - memcached_connections_overhead)` é a memória do nó utilizável para dados. Para obter mais informações, consulte [Sobrecarga de conexões do Memcached](#).

Considerações sobre atualização ao trabalhar com clusters autoprojados

Note

As considerações a seguir só se aplicam ao atualizar clusters autoprojados. Elas não se aplicam ao ElastiCache Sem Servidor.

Ao atualizar um cluster autoprojado, 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.
- Como o mecanismo Memcached não oferece suporte para persistência, as atualizações de versão do mecanismo Memcached são sempre um processo disruptivo que limpa todos os dados do cache no cluster.

Como atualizar as versões dos mecanismos

Para iniciar as atualizações de versão no seu cluster, modifique-o e especifique uma versão mais recente do mecanismo. Faça isso usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache:

- Para usar o AWS Management Console, consulte [Modifying clusters through the console](#).
- Para usar o AWS CLI, consulte [Modifying clusters with the CLI](#).
- Para usar a API do ElastiCache, consulte [Modifying clusters through the API](#).

Como atualizar as versões dos mecanismos

Para iniciar as atualizações de versão no seu cluster, modifique-o e especifique uma versão mais recente do mecanismo. Faça isso usando o console do ElastiCache, a AWS CLI ou a API do ElastiCache:

- Para usar a AWS Management Console, consulte - [Usando o AWS Management Console](#).
- Para usar a AWS CLI, consulte [Usando o AWS CLI](#).
- Para usar a API do ElastiCache, consulte [Usando a ElastiCache API](#).

Práticas recomendadas e estratégias de armazenamento em cache do ElastiCache

Abaixo, é possível encontrar práticas recomendadas para o Amazon ElastiCache. Seguir essas práticas melhora o desempenho e aumenta a confiabilidade do cache.

Tópicos

- [Práticas recomendadas com clientes do Memcached](#)
- [Comandos do Memcached compatíveis](#)
- [Estratégias de armazenamento em cache](#)

Práticas recomendadas com clientes do Memcached

Para saber mais sobre as práticas recomendadas para interagir com recursos do ElastiCache com bibliotecas cliente do Memcached de código-fonte aberto, consulte os tópicos abaixo.

Tópicos

- [Configuração do seu cliente do ElastiCache para balanceamento de carga eficiente](#)
- [Exemplos de cliente de IPv6](#)

Configuração do seu cliente do ElastiCache para balanceamento de carga eficiente

Note

Esta seção se aplica a clusters do Memcached multinó autoprojeto.

Para usar efetivamente vários nós do Memcached do ElastiCache, você precisa ser capaz de espalhar suas chaves de cache entre os nós. Uma maneira simples de balancear a carga de um cluster com n nós é calcular o hash da chave do objeto e modificar o resultado por $n - \text{hash}(\text{key}) \bmod n$. O valor resultante (de 0 a $n-1$) é o número do nó em que você coloca o objeto.

Essa abordagem é simples e funcionará bem desde que o número de nós (n) seja constante. No entanto, sempre que você adiciona ou remove um nó do cluster, o número de chaves que precisam ser movidas é $(n - 1) / n$ (em que n é o novo número de nós). Assim, essa abordagem resulta em um grande número de chaves movidas, o que se converte em um grande número de perdas no cache inicial, especialmente porque o número de nós fica grande. A escala de 1 para 2 nós resulta na movimentação de $(2-1)/2$ (50%) das chaves, o melhor caso. A escala de 9 a 10 nós resulta na movimentação de $(10-1)/10$ (90%) das chaves. Se você estiver aumentando a escala devido a um pico no tráfego, não convém ter um grande número de perdas no cache. Um grande número de perdas no cache resulta em acertos no banco de dados, que já está sobrecarregado devido ao pico no tráfego.

A solução para esse dilema é o hashing consistente. O hashing consistente usa um algoritmo, de modo que, sempre que um nó é adicionado ou removido a/de um cluster, o número de chaves que devem ser movidas é de aproximadamente $1/n$ (onde n é o novo número de nós). A escala de 1 para 2 nós resulta na movimentação de $1/2$ (50%) das chaves, o pior caso. Escalar de 9 a 10 nós resulta na movimentação de $1/10$ (10%) das chaves.

Como usuário, você controla qual algoritmo de hashing é usado para clusters de vários nós. Recomendamos configurar os clientes para usar o hashing consistente. Felizmente, existem muitas bibliotecas de cliente Memcached na maioria das linguagens populares que implementam o hashing consistente. Verifique a documentação da biblioteca que você está usando para ver se ela oferece suporte para hashing consistente e como implementá-lo.

Se você estiver trabalhando em Java, PHP ou .NET, recomendamos o uso de uma das bibliotecas de cliente do Amazon ElastiCache.

Hash consistente usando o Java

O cliente Java Memcached do ElastiCache baseia-se no cliente Java spymemcached de código-fonte aberto, que possui capacidades de hashing consistente incorporadas. A biblioteca inclui uma classe `KetamaConnectionFactory` que implementa o hashing consistente. Por padrão, o hashing consistente está desativado em spymemcached.

Para obter mais informações, consulte a documentação do `KetamaConnectionFactory` em [KetamaConnectionFactory](#).

Hash consistente usando o PHP

O cliente PHP Memcached do ElastiCache é um wrapper ao redor da biblioteca PHP Memcached interna. Por padrão, o hashing consistente é desativado pela biblioteca PHP Memcached.

Use o código a seguir para ativar o hashing consistente.

```
$m = new Memcached();  
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Além do código anterior, recomendamos que você também ative `memcached.sess_consistent_hash` no seu arquivo `php.ini`.

Para obter mais informações, consulte a documentação de configuração de tempo de execução do PHP Memcached em <http://php.net/manual/en/memcached.configuration.php>. Observe especificamente o parâmetro `memcached.sess_consistent_hash`.

Hash consistente usando o .NET

O cliente .NET Memcached do ElastiCache é um wrapper ao redor do Enyim Memcached. Por padrão, o hashing consistente é ativado pelo cliente Enyim Memcached.

Para obter mais informações, consulte a documentação de `memcached/locator` em <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator>.

Exemplos de cliente de IPv6

Note

Esta seção se aplica a clusters do Memcached autoprojeto.

ElastiCache é compatível com o Memcached de código aberto. Isso significa que os clientes de código aberto do Memcached que oferecem suporte a conexões IPv6 devem ser capazes de se conectar ao IPv6 habilitado para clusters do Memcached. Além disso, os seguintes clientes foram especificamente validados para funcionar com todas as configurações de tipo de rede compatíveis:

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

Cientes validados:

- [AWS ElastiCache Cliente de cluster Memcached para PHP — Versão *3.6.2](#)
- [AWS ElastiCache Cluster Client Memcached para Java](#) — Master mais recente no Github

Configurar um protocolo preferencial para clusters de pilha dupla

Para clusters Memcached, você pode controlar o protocolo que os clientes usarão para se conectar aos nós no cluster com o parâmetro de descoberta de IP. O parâmetro de descoberta de IP pode ser definido como IPv4 ou IPv6.

O parâmetro de descoberta de IP controla o protocolo IP usado na saída do cluster `config get`. O que, por sua vez, determinará o protocolo IP usado pelos clientes que oferecem suporte à descoberta automática ElastiCache para clusters do Memcached.

Alterar a descoberta de IP não resultará em nenhum tempo de inatividade para os clientes conectados. No entanto, as alterações levarão algum tempo para se propagar.

Monitore a saída de `getAvailableNodeEndpoints` para Java e para Php monitore a saída de `getServerList`. Depois que a saída dessas funções relata IPs de todos os nós do cluster que usam o protocolo atualizado, as alterações terminam de se propagar.

Exemplo de Java

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

```

Class targetProtocolType = InetAddress.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

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

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
    } catch (UnknownHostException ignored) {}
    return false;
})));

```

Exemplo de PHP:

```

$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
    # The PHP memcached client only updates the server list if the polling interval has
    expired and a
    # command is sent
    $client->get('test');

    $nodes = $client->getServerList();

    sleep(1);
    $target_ips_count = 0;

    // For IPv4 use FILTER_FLAG_IPV4
    $target_ips_count = count(array_filter($nodes, function($node) { return
    filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));
} while (count($nodes) !== $target_ips_count);

```

Qualquer conexão de cliente existente que tenha sido criada antes da atualização da descoberta de IP ainda será conectada usando o protocolo antigo. Todos os clientes validados se reconectarão automaticamente ao cluster usando o novo protocolo IP assim que as alterações forem detectadas na saída dos comandos de descoberta do cluster. No entanto, isso depende da implementação do cliente.

Clusters de ElastiCache pilha dupla habilitados para TLS

Quando o TLS está habilitado para ElastiCache clusters, as funções de descoberta de cluster 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.

Clientes 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.

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 `172.31.0.0/16`, 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
```


Comandos do Memcached compatíveis

O ElastiCache Sem Servidor para Memcached dá suporte a todos os [comandos](#) memcached no memcached 1.6 de código-fonte aberto, exceto os seguintes:

- Como as conexões cliente exigem TLS, o protocolo UDP não é compatível.
- O protocolo binário não é compatível, pois está oficialmente [obsoleto](#) no memcached 1.6.
- Os comandos GET/GETS são limitados a 16 KB para evitar possíveis ataques de DoS ao servidor com a busca de um grande número de chaves.
- O comando `flush_all` atrasado será rejeitado com `CLIENT_ERROR`.
- Comandos que configuram o mecanismo ou revelam informações internas sobre o estado ou os logs do mecanismo não são compatíveis, como:
 - Para comando `STATS`, só `stats` e `stats reset` são compatíveis. Outras variações vão retornar `ERROR`
 - `lru / lru_crawler`: modificação das configurações de crawler LRU e LRU
 - `watch`: observa os logs de servidor do memcached
 - `verbosity`: configura o nível de log do servidor

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 Memcached especifica este valor em segundos. Quando uma aplicação tenta ler uma chave expirada, isso é tratado como se a chave não fosse encontrada. O banco de dados é consultado para a chave e o cache é atualizado. Essa abordagem não garante que um valor não esteja obsoleto. Contudo, isso impede que os dados fiquem obsoletos demais e exige que os valores no cache sejam ocasionalmente atualizados a partir do banco de dados.

Para obter mais informações, consulte o [Memcached 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 Memcached](#)

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

- [Gerenciamento da manutenção](#)
- [Configuração de parâmetros do mecanismo usando grupos de parâmetros](#)

Gerenciamento da manutenção

Cada cluster 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, o ElastiCache atribuirá uma janela de manutenção de 60 minutos dentro da janela de manutenção da sua região em um dia da semana escolhido aleatoriamente.

A janela de manutenção de 60 minutos é escolhida aleatoriamente entre um período de 8 horas por região. A tabela a seguir lista os blocos de tempo de cada região dos quais as janelas de manutenção padrão são atribuídas. Você pode escolher uma janela de manutenção preferida fora do bloco de janelas de manutenção da região.

Código da região	Nome da região	Janela de manutenção da região
ap-northeast-1	Região Ásia-Pacífico (Tóquio)	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)

Código da região	Nome da região	Janela de manutenção da região
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)
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

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 para especificar um intervalo 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
- [Modificando um cluster ElastiCache](#) : Alteração da janela de manutenção de um cluster

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 Memcached](#).

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 do tipo de nó do Memcached](#).

Note

Para obter uma lista de parâmetros específicos do Memcached, consulte [Parâmetros específicos de Memcached](#).

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](#)

Gerenciamento de parâmetros

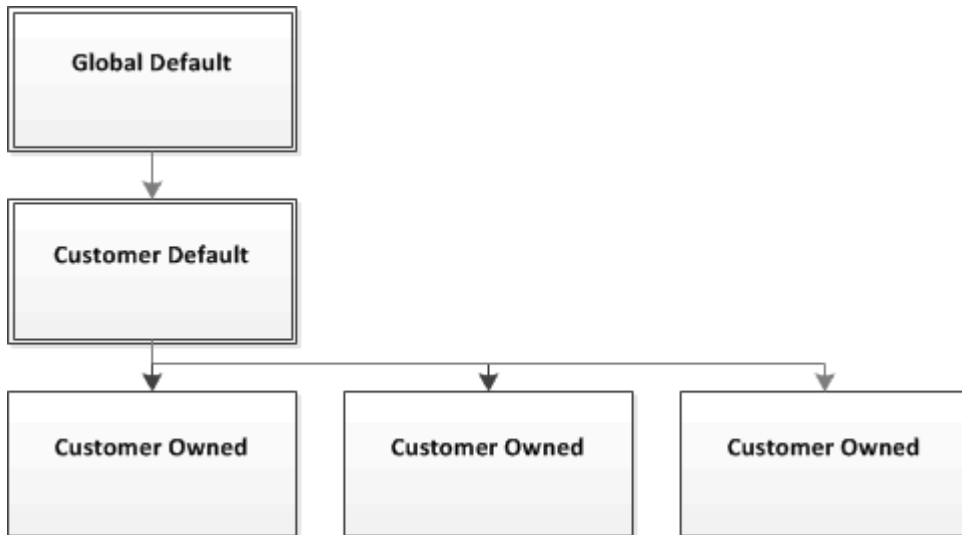
Os parâmetros são agrupados em parameter groups nomeados para facilitar o gerenciamento de parâmetros. Um parameter group representa uma combinação de valores específicos para os parâmetros que são transmitidos ao software do mecanismo durante a inicialização. Esses valores determinam como o processo 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 Memcached versão 1.4.8, você só poderá usar parameter groups, padrão ou personalizados, da família Memcached 1.4.
- 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 os parâmetros de um cluster, a alteração é aplicada ao cluster imediatamente. Isso é verdadeiro se você alterar o próprio grupo de parâmetro do cluster ou um valor do parâmetro dentro do grupo do parâmetro do cluster. Para determinar quando uma alteração de parâmetro específica é aplicada, consulte a coluna *As alterações terão efeito nas tabelas para* [Parâmetros específicos do Memcached](#). Para obter informações sobre como reinicializar nós de um cluster, consulte [Rebooting clusters](#).

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 Memcached](#).

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 memcached1.4 , 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 myMem14 usando a família memcached1.4 como modelo.

Para Linux, macOS ou Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --cache-parameter-group-family memcached1.4 \  
  --description "My first parameter group"
```

Para Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --cache-parameter-group-family memcached1.4 ^  
  --description "My first parameter group"
```

A saída desse comando deve ser semelhante a esta.

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myMem14",  
    "CacheParameterGroupFamily": "memcached1.4",  
    "Description": "My first parameter group"  
  }  
}
```

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, `memcached1.4`.
- **Description:** uma descrição fornecida pelo usuário para o grupo de parâmetros.

Example

O exemplo a seguir cria um parameter group chamado `myMem14` usando a família `memcached1.4` como modelo.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheParameterGroup  
&CacheParameterGroupFamily=memcached1.4  
&CacheParameterGroupName=myMem14  
&Description=My%20first%20parameter%20group  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

A resposta dessa ação deve ser algo semelhante ao seguinte.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <CreateCacheParameterGroupResult>  
    <CacheParameterGroup>  
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>  
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>  
      <Description>My first parameter group</Description>  
    </CacheParameterGroup>  
  </CreateCacheParameterGroupResult>
```

```
<ResponseMetadata>
  <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
</ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

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 `myMem14`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myMem14
```

Para Windows:

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myMem14
```

A saída desse comando será algo assim, listando o nome, a família e a descrição do parameter group.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myMem14",
      "CacheParameterGroupFamily": "memcached1.4",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

O código de exemplo a seguir lista 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 o parameter group `myMem14`.

```

https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myMem14
&SignatureVersion=4
&SignatureMethod=HmacSHA256

```

```
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

A resposta dessa ação parecerá algo semelhante ao seguinte, listando o nome, a família e a descrição de cada parameter group.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

O código de exemplo a seguir lista até 10 parameter groups.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&MaxRecords=10
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

A resposta dessa ação será semelhante ao seguinte, listagem do nome, família, descrição e, no caso do redis5.6, se o grupo de parâmetros pertence a um datastore (isGlobal), para cada grupo de parâmetros.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
```

```
<DescribeCacheParameterGroupsResult>
  <CacheParameterGroups>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
      <Description>My custom Redis 2.8 parameter group</Description>
    </CacheParameterGroup>
    <CacheParameterGroup>
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
      <Description>My custom Memcached 1.4 parameter group</Description>
    </CacheParameterGroup>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
      <Description>My custom redis 5.6 parameter group</Description>
      <isGlobal>yes</isGlobal>
    </CacheParameterGroup>
  </CacheParameterGroups>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
  <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

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 `myMem14`.

Para Linux, macOS ou Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myMem14
```

Para Windows:

```
aws elasticache describe-cache-parameters ^
```



```
--cache-parameter-group-name myMem14
```

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 myMem14.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameters  
&CacheParameterGroupName=myMem14  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

A resposta dessa ação será semelhante a esta. Essa resposta foi truncada.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParametersResult>  
    <CacheClusterClassSpecificParameters>  
      <CacheNodeTypeSpecificParameter>  
        <DataType>integer</DataType>  
        <Source>system</Source>  
        <IsModifiable>>false</IsModifiable>  
        <Description>The maximum configurable amount of memory to use to store items,  
in megabytes.</Description>  
        <CacheNodeTypeSpecificValues>  
          <CacheNodeTypeSpecificValue>  
            <Value>1000</Value>  
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>  
          <CacheNodeTypeSpecificValue>  
            <Value>6000</Value>  
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>
```

```
<CacheNodeTypeSpecificValue>
  <Value>7100</Value>
  <CacheClusterClass>cache.m1.large</CacheClusterClass>
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>1300</Value>
  <CacheClusterClass>cache.m1.small</CacheClusterClass>
</CacheNodeTypeSpecificValue>

...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 Memcached](#).

Modificação de um grupo de parâmetros (console)

O procedimento a seguir mostra como alterar o valor do parâmetro `binding_protocol` 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 a lista para ver todos os parâmetros.

4. Para modificar um ou mais parâmetros, escolha Edit Parameters.
5. Na tela Edit Parameter Group:, percorra usando as setas esquerda e direita até encontrar o parâmetro `binding_protocol`. Em seguida, digite `ascii` na coluna Value.
6. Na tela Edit Parameter Group:, percorra usando as setas esquerda e direita até encontrar o parâmetro `cluster-enabled`. Em seguida, digite `yes` na coluna Value.
7. Escolha Save Changes (Salvar alterações).
8. Para localizar o parâmetro que você alterou, consulte [Parâmetros específicos do Memcached](#). Se alterações no parâmetro ocorrerem depois da reinicialização (After restart), reinicialize todos os clusters que usam esse parameter group. Para obter mais informações, consulte [Rebooting clusters](#).

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 Memcached](#)

O código de exemplo a seguir define o valor de dois parâmetros, `chunk_size` e `chunk_size_growth_fact` no parameter group `myMem14`.

Para Linux, macOS ou Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --parameter-name-values \  
    ParameterName=chunk_size,ParameterValue=96 \  
  --parameter-name-values
```

```
ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Para Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --parameter-name-values ^  
    ParameterName=chunk_size,ParameterValue=96 ^  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

A saída desse comando será semelhante a esta.

```
{  
  "CacheParameterGroupName": "myMem14"  
}
```

Para obter mais informações, consulte [modify-cache-parameter-group](#).

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 Memcached](#)

O código de exemplo a seguir define o valor de dois parâmetros, `chunk_size` e `chunk_size_growth_fact` no parameter group `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheParameterGroup  
&CacheParameterGroupName=myMem14  
&ParameterNameValues.member.1.ParameterName=chunk_size  
&ParameterNameValues.member.1.ParameterValue=96  
&ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact  
&ParameterNameValues.member.2.ParameterValue=1.5  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z
```

```
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

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

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 myMem14
```

Para Windows:

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

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 `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myMem14  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

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 1511 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: booliano</p> <p>Modificável: sim</p> <p>Valores permitidos: 0,1</p> <p>As alterações terão efeito: na execução</p>	Armazena números da resposta VALUE, dentro de um item, usando até 24 bytes. Pequena desaceleração para 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: booliano</p> <p>Modificável: sim</p> <p>Valores permitidos: 0,1</p> <p>As alterações terão efeito: na execução</p>	Não é mais compatível nesta versão.
<code>modern</code>	Padrão: 1	

Nome	Detalhes	Descrição
	<p>Tipo: booliano</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: booliano</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: booliano</p> <p>Modificável: sim</p> <p>As alterações terão efeito: após a reinicialização</p> <div data-bbox="651 541 971 1381" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> 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 1793" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 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>	Padrão: 0 (false) Tipo: booliano Modificável: sim As alterações terão efeito: após a reinicialização	Se 1 (true), quando não há mais memória para armazenar itens, o Memcached retornará um erro em vez de remover itens.
<code>large_memory_pages</code>	Padrão: 0 (false) Tipo: booliano Modificável: não	Se 1 (true), o ElastiCache tentará usar páginas de memória grandes.
<code>lock_down_paged_memory</code>	Padrão: 0 (false) Tipo: booliano Modificável: não	Se 1 (true), o ElastiCache bloqueará toda a memória paginada.
<code>max_item_size</code>	Padrão: 1048576 Tipo: inteiro Modificável: sim As alterações terão efeito: após a reinicialização	O tamanho, em bytes, do maior item que pode ser armazenado no cluster.
<code>max_simultaneous_connections</code>	Padrão: 65000 Tipo: inteiro Modificável: não	O número máximo de conexões simultâneas.

Nome	Detalhes	Descrição
<code>maximize_core_file_limit</code>	<p>Padrão: 0 (false)</p> <p>Tipo: booleano</p> <p>Permite modificação:</p> <p>As alterações terão efeito: após a reinicialização</p>	Se 1 (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).

Dimensionamento ElastiCache para Memcached

Dimensionamento ElastiCache para Memcached

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 usando a lógica LRU. 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 Memcached.

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-escalonamento 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-escalonamento

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 autoprotetidos pelo Memcached

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.

Escalar clusters Memcached

Ação	Tópico
Aumento de escala	Adição de nós a um cluster
Redução de escala	Exclusão de nós de um cluster
Alteração nos tipos de nó	Escalabilidade vertical do Memcached

Os clusters do Memcached são compostos de 1 a 60 nós. Escalar um cluster Memcached é tão fácil quanto adicionar ou remover nós do cluster.

Se você precisar de mais de 60 nós em um cluster Memcached ou de mais de 300 nós no total em uma AWS região, preencha o formulário de solicitação de aumento de ElastiCache limite em <https://aws.amazon.com/contact-us/elasticache-node-limit-request/>.

Como você pode particionar seus dados em todos os nós em um cluster Memcached, raramente é necessário expandir para um tipo de nó com mais memória. No entanto, como o mecanismo Memcached não mantém a persistência de dados, se você dimensionar um tipo de nó diferente, seu novo cluster Memcached começará vazio, a menos que seu aplicativo o preencha.

Tópicos

- [Escalar o Memcached horizontalmente](#)
- [Escalabilidade vertical do Memcached](#)

Escalar o Memcached horizontalmente

O mecanismo Memcached oferece suporte para particionar seus dados em vários nós. Por isso, os clusters Memcached escalam horizontalmente com facilidade. Um cluster Memcached pode ter de 1 a 60 nós. Para dimensionar horizontalmente seu cluster Memcached, basta adicionar ou remover nós.

Se você precisar de mais de 60 nós em um cluster Memcached ou de mais de 300 nós no total em uma AWS região, preencha o formulário de solicitação de aumento de ElastiCache limite em <https://aws.amazon.com/contact-us/elasticache-node-limit-request/>.

Os tópicos a seguir detalham como escalar seu cluster Memcached inserindo ou removendo nós.

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

Cada vez que você altera o número de nós no seu cluster Memcached, você deve remapear pelo menos alguns dos seus espaços de chaves para que ele seja mapeado para o nó correto. Para obter informações mais detalhadas sobre o balanceamento de carga do cluster Memcached, consulte [Configuração do seu cliente do ElastiCache para balanceamento de carga eficiente](#).

Se você usar a descoberta automática no seu cluster Memcached, não precisará alterar os endpoints em seu aplicativo à medida que você adiciona ou remove nós. Para obter mais informações sobre a descoberta automática, consulte [Identificar automaticamente os nós no seu cluster](#) Se você não usar

a descoberta automática, cada vez que você alterar o número de nós no seu cluster Memcached, deverá atualizar os endpoints no seu aplicativo.

Escalabilidade vertical do Memcached

Quando você expande ou reduz seu cluster Memcached, deve criar um novo cluster. Os clusters Memcached sempre começam vazios, a menos que seu aplicativo o preencha.

Important

Se você estiver dimensionando para um tipo de nó menor, certifique-se de que o tipo de nó menor seja adequado para seus dados e despesas gerais. Para obter mais informações, consulte [Select cache node size](#).

Tópicos

- [Escalabilidade vertical do Memcached \(console\)](#)
- [Escalabilidade vertical do Memcached \(AWS CLI\)](#)
- [Escalando o Memcached verticalmente \(API\) ElastiCache](#)

Escalabilidade vertical do Memcached (console)

O procedimento a seguir mostra como escalar seu cluster verticalmente usando o ElastiCache console.

Para escalar um cluster do Memcached verticalmente (console)

1. Crie um novo cluster com o novo tipo de nó. Para ter mais informações, consulte [Criação de um cluster do Memcached \(console\)](#).
2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para ter mais informações, consulte [Localização de endpoints de um cluster \(console\)](#).
3. Exclua o cluster antigo. Para obter mais informações, consulte [Deleting a new node in Memcached](#).

Escalabilidade vertical do Memcached (AWS CLI)

O procedimento a seguir demonstra passo a passo como escalonar seu cluster de cache Memcached verticalmente usando a AWS CLI.

Para escalar um cluster de cache Memcached verticalmente (AWS CLI)

1. Crie um novo cluster de cache com o novo tipo de nó. Para obter mais informações, consulte [Creating clusters with the CLI](#).
2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster. Para ter mais informações, consulte [Localizar endpoints \(AWS CLI\)](#).
3. Exclua o cluster de cache antigo. Para ter mais informações, consulte [Usar a AWS CLI](#).

Escalando o Memcached verticalmente (API) ElastiCache

O procedimento a seguir mostra como escalar verticalmente seu cluster de cache do Memcached usando a API. ElastiCache

Para escalar verticalmente um cluster de cache do Memcached (API) ElastiCache

1. Crie um novo cluster de cache com o novo tipo de nó. Para obter mais informações, consulte [Criação de um cluster \(ElastiCache API\)](#).
2. No seu aplicativo, atualize os endpoints para os endpoints do novo cluster de cache. Para ter mais informações, consulte [Localização de endpoints \(API do ElastiCache\)](#).
3. Exclua o cluster de cache antigo. Para ter mais informações, consulte [Uso da API do ElastiCache](#).

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 você tem muitos recursos do mesmo tipo. É possível identificar rapidamente um recurso específico baseado nas tags que você atribuiu a ele. Este tópico descreve tags e mostra a você como criá-los.

Warning

Como uma prática recomendada, sugerimos que você não inclua dados confidenciais nas suas tags.

Conceitos Básicos de Tags

Uma tag é um rótulo atribuído a um recurso da AWS. Cada tag consiste em uma chave e um valor opcional, ambos definidos por você. As tags permitem categorizar seus recursos da AWS de diferentes formas, por exemplo, por finalidade ou por proprietário. Por exemplo, você pode definir um conjunto de tags para os clusters do ElastiCache sua conta que ajudem a rastrear o proprietário e o grupo de usuários de cada instância.

Recomendamos que você desenvolva um conjunto de chave de tags que atenda suas necessidades para cada tipo de recurso. Usar um conjunto consistente de chaves de tags facilita para você gerenciar seus recursos da . É possível pesquisar e filtrar os recursos de acordo com as tags que adicionar. Para obter mais informações sobre como implementar uma estratégia eficaz de marcação de recursos, consulte o [whitepaper da AWS, Tagging Best Practices](#) (Práticas recomendadas de marcação).

As tags não têm significado semântico no ElastiCache e são interpretadas estritamente como uma string de caracteres. Além disso, as tags não são automaticamente atribuídas aos seus recursos. É possível editar chaves de tags e valores, e é possível remover as tags de um recurso a qualquer momento. É possível definir o valor de uma tag como `null`. Ao adicionar uma tag que tenha a mesma chave de uma tag existente nesse recurso, o novo valor substituirá o antigo. Se você excluir um recurso, todas as tags do recurso também serão excluídas. Além disso, se você adicionar ou excluir tags em um grupo de replicação, todos os nós nesse grupo de replicação também terão suas tags adicionadas ou removidas.

Você pode trabalhar com tags usando o AWS Management Console, a AWS CLI e a API do ElastiCache.

Se você estiver usando o IAM, pode controlar quais usuários na sua conta da AWS têm permissão para criar, editar ou excluir tags. Para obter mais informações, consulte [Permissões em nível de recurso](#).

Recursos que podem ser marcados

Você pode 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`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `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
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim
Sim	Sim

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](#).

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 memcached \  
  --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Adicionar tags a um cache sem servidor

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Criação de um cluster de cache usando tags.

```
aws elasticache create-cache-cluster \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

```
--cluster-id testing-tags \
--cluster-description cluster-test \
--cache-subnet-group-name test \
--cache-node-type cache.t2.micro \
--engine memcached \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

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. 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 clusters do Memcached. Ao adicionar, listar, modificar, copiar ou remover uma tag, a operação é aplicada somente ao cluster 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>`

Memcached: as tags são aplicadas somente aos clusters.

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.

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`

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 cluster Memcached `my-cluster` na região `us-west-2`.

Para Linux, macOS ou Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Para Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

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 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 \
  --tags Key=Service,Value=elasticache \
  Key=Region,Value=us-west-2
```

Para Windows:

```
aws elasticache add-tags-to-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
  --tags Key=Service,Value=elasticache ^
```

```
Key=Region,Value=us-west-2
```

A saída dessa operação será semelhante a uma lista de todas as tags no recurso após a operação, conforme mostrado a seguir.

```
{
  "TagList": [
    {
      "Value": "elasticache",
      "Key": "Service"
    },
    {
      "Value": "us-west-2",
      "Key": "Region"
    }
  ]
}
```

Para obter mais informações, consulte a AWS CLI sobre o ElastiCache [add-tags-to-resource](#).

Você também pode usar a AWS CLI para adicionar tags a um cluster ao criar um novo cluster usando a operação [create-cache-cluster](#). Não é possível adicionar tags ao criar um cluster usando o console de gerenciamento do ElastiCache. Depois que o cluster é criado, você pode usar o console para adicionar tags ao cluster.

Modificação de tags usando a AWS CLI

Você pode usar a AWS CLI para modificar as tags em um cluster do ElastiCache for Memcached.

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 de tags usando a AWS CLI

Você pode usar a AWS CLI para remover tags de um cluster do ElastiCache for Memcached existente usando a operação [remove-tags-from-resource](#).

O código a seguir usa a AWS CLI para remover as etiquetas com as chaves `Service` e `Region` do cluster `my-cluster` na região `us-west-2`.

Para Linux, macOS ou Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \  
  --tag-keys PM Service
```

Para Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
  --tag-keys PM Service
```

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` na região `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListTagsForResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

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` 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  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Tags.member.1.Key=Service  
&Tags.member.1.Value=elasticache  
&Tags.member.2.Key=Region
```



```
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

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

Modificação de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para modificar as tags em um cluster do ElastiCache.

Para modificar o valor de uma tag:

- Use a operação [AddTagsToResource](#) para adicionar uma nova tag e um valor ou para alterar o valor de uma tag existente.
- Use [RemoveTagsFromResource](#) para remover tags do recurso.

A saída de qualquer operação será uma lista de tags e seus valores no recurso especificado.

Use [RemoveTagsFromResource](#) para remover tags do recurso.

Remoção de tags usando a API do ElastiCache

Você pode usar a API do ElastiCache para remover tags de um cluster do ElastiCache for Memcached existente usando a operação [RemoveTagsFromResource](#).

O código a seguir usa a API do ElastiCache para remover as tags com as chaves `Service` e `Region` do 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
&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] 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 LRU 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](#)
 - [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]:

- [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]:

- [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 `CurConnections` 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](#)
 - [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 autoprojeto 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) ou 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` `EngineCPUUtilization` e `EngineCPUUtilization` fornece a utilização da CPU dedicada ao processo Redis e o uso em todas `CPUUtilization` as vCPUs. Os nós com mais de uma vCPU geralmente têm valores diferentes para `CPUUtilization` e `EngineCPUUtilization`, sendo o segundo comumente maior. Um `EngineCPUUtilization` alto pode ser causado por um número elevado de solicitações ou operações complexas que levem uma quantidade significativa de tempo da CPU para ser concluída. Você pode identificar ambos com o seguinte:
 - **Número elevado de solicitações:** verifique se há aumentos em outras métricas que correspondam ao padrão `EngineCPUUtilization`. As métricas úteis são:
 - **CacheHits e CacheMisses:** o número de solicitações ou solicitações bem-sucedidas que não encontraram um item válido no cache. Se a proporção de erros em comparação com acertos for alta, a aplicação está desperdiçando tempo e recursos com solicitações infrutíferas.
 - **SetTypeCmds e GetTypeCmds:** Essas métricas correlacionadas com `EngineCPUUtilization` podem ajudar a entender se a carga é significativamente maior para solicitações de gravação, medida por `SetTypeCmds`, ou leituras, medido por `GetTypeCmds`. Se a carga for predominantemente de leituras, o uso de várias réplicas de leitura pode equilibrar as solicitações em vários nós e poupar o primário para gravações. Em clusters 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 atuam sobre vários valores, chaves ou tipos de dados deve ser feito com cuidado. As conexões podem ser bloqueadas ou terminadas dependendo do número de parâmetros ou do tamanho de seus valores de entrada ou saída.

Um exemplo notório é o comando KEYS. Ele varre todo o keyspace procurando por um determinado padrão e bloqueia a execução de outros comandos durante sua execução. O Redis usa a notação "Big O" para descrever sua complexidade de comandos.

O comando Keys tem complexidade de tempo $O(N)$, sendo N o número de chaves no banco de dados. Portanto, quanto maior o número de chaves, mais lento será o comando. KEYS pode causar problemas de maneiras diferentes: se nenhum padrão de pesquisa for usado, o comando retornará todos os nomes de chaves disponíveis. Em bancos de dados com milhares ou milhões de itens, uma enorme saída será criada e inundará os buffers de rede.

Se um padrão de pesquisa for usado, somente as chaves correspondentes ao padrão retornarão ao cliente. No entanto, o mecanismo ainda varrerá todo o keyspace procurando por ele, e o tempo para concluir o comando será o mesmo.

Uma alternativa a KEYS é o comando SCAN. Ele itera sobre o keyspace e limita as iterações em um número específico de itens, evitando bloqueios prolongados no mecanismo.

A varredura tem o parâmetro COUNT, usado para definir o tamanho dos blocos de iteração. O valor padrão é 10 (10 itens por iteração).

Dependendo do número de itens no banco de dados, blocos com pequenos valores de COUNT irão exigir mais iterações para concluir uma verificação completa, e valores maiores manterão o mecanismo ocupado por mais tempo em cada iteração. Enquanto pequenos valores de

contagem farão SCAN mais lento em grandes bancos de dados, valores maiores podem causar os mesmos problemas mencionados para KEYS.

Como exemplo, a execução do comando SCAN com valor de contagem de 10 vai requer 100.000 repetições em um banco de dados com 1 milhão de chaves. Se o tempo médio de ida e volta da rede for de 0,5 milissegundos, aproximadamente 50.000 milissegundos (50 segundos) serão gastos transferindo solicitações.

Por outro lado, se o valor da contagem fosse 100.000, uma única iteração seria necessária e apenas 0,5 ms seriam gastos transferindo-a. No entanto, o mecanismo seria totalmente bloqueado para outras operações até que o comando terminasse de varrer todo o keyspace.

Além de KEYS, vários outros comandos são potencialmente prejudiciais se não forem usados corretamente. Para ver uma lista de todos os comandos e suas respectivas complexidades de tempo, acesse <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 keyspace serão encerrados automaticamente após o período de 5 segundos. Para evitar a corrupção de dados e inconsistências, o nó fará failover se a execução do script não tiver sido concluída em 5 segundos e tiver qualquer gravação durante sua execução. [Transações](#) são a alternativa para garantir a consistência de múltiplas modificações de chaves relacionadas no Redis. 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 `ConnConnections` 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`, `ttdl`, `type`, `del`, `expire`, `expireat`, `move`, `persist`, `pexpire`;
 - **ListBasedCmdsLatency:** `lindex`, `len`, `lrange`, `lrangebypop`, `linsert`, `pop`, `push`, `pushx`, `lrem`, `ltrim`, `rop`, `roplpush`, `rpush`, `rpushx`;
 - **PubSubBasedCmdsLatency:** `punsubscribe`, `publike`, `pubsub`, `cancel`, `unsubscribe`, `cancel`;
 - **SetBasedCmdsLatency:** `scard`, `sdiff`, `sinter`, `sismember`, `smembers`, `srandmember`, `union`, `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`,

- zrangebylex, zlexcount, zadd, zincrby, zinterstore, zrem, zremrangebyrank, zremrangebyscore, zunionstore, zremrangebylex, zpopmax, zpopmin, bzipopmin, bzipopmax;
- **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;
 - `conntrack_allowance_exceeded`: número de pacotes descartados devido a [limites de monitoramento de conexão](#) de grupos de segurança. Novas conexões falharão quando esse limite estiver saturado;
 - `linklocal_allowance_exceeded`: número de pacotes descartados devido a solicitações excessivas de metadados de instâncias, NTP via VPC DNS. O limite é de 1024 pacotes por segundo para todos os serviços;

- `pps_allowance_exceeded`: número de pacotes descartados devido à proporção excessiva de pacotes por segundo. O limite de PPS pode ser atingido quando o tráfego de rede consistir em milhares ou milhões de solicitações muito pequenas por segundo. ElastiCache o tráfego pode ser otimizado para fazer melhor uso dos pacotes de rede por meio de pipelines ou comandos que realizam várias operações ao mesmo tempo, em vez de. MGET GET

Dissecação do tempo necessário para concluir uma única solicitação

- On the network: Tcpcdump and Wireshark (tshark na linha de comando) são ferramentas úteis para entender quanto tempo a solicitação levou para percorrer a rede, acionar o ElastiCache motor e obter uma devolução. O exemplo a seguir destaca uma única solicitação criada com o seguinte comando:

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

Paralelamente ao comando acima, tcpcdump estava em execução e retornou:

```
$ sudo tcpcdump -i any -nn port 6379 -tt
tcpcdump: 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"], 0x7ffffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
    (+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
    (+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3

```

```

1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
IPPROTO_IP) = 3
1609430221.717419
    (+ 0.004126) recvfrom(3, "\362|
\201\200\0\1\0\2\0\0\0\0\0\0\notls20201224\6tihew"... , 2048, 0, {sa_family=AF_INET,
sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
"\204\207\201\200\0\1\0\1\0\0\0\0\0\notls20201224\6tihew"... ,
65536, 0, {sa_family=AF_INET, sin_port=htons(53),
sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
[128]) = -1 ENOTSOCK
(Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
[128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
    (+ 0.003569) +++ exited with 0 +++

```

No exemplo acima, o comando levou um pouco mais de 54 milissegundos para ser concluído (752110 - 697712 = 54398 microssegundos).

Um tempo significativo, aproximadamente 20 ms, foi levado para instanciar nc e fazer a resolução do nome (de 697712 a 717890), depois disso, 2ms foram necessários para criar o soquete de TCP (745659 a 747858), e 0,4 ms (747858 a 748330) para enviar e receber a resposta para a solicitação.

Segurança no Amazon ElastiCache

A segurança na nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isto como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem:** a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [compliance programsAWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao Amazon ElastiCache, consulte [Serviços da AWS no escopo por programa de conformidade](#).
- **Segurança na nuvem:** sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon ElastiCache. Os tópicos a seguir mostram como configurar o Amazon ElastiCache para atender aos seus objetivos de segurança e de conformidade. Saiba também como usar outros serviços da AWS que ajudam você a monitorar e proteger os recursos do Amazon ElastiCache.

Tópicos

- [Proteção de dados no Amazon ElastiCache](#)
- [Privacidade do tráfego entre redes](#)
- [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](#)

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 Memcached também oferece recursos de criptografia para dados em caches que estiverem executando as versões 1.6.12 ou posterior do Memcached.

- 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.

Tópicos

- [Criptografia em trânsito \(TLS\) do ElastiCache](#)
- [Criptografia em repouso no ElastiCache](#)

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 autoprojeto, você pode habilitar a criptografia em trânsito em um cluster de cache definindo o parâmetro como `TransitEncryptionEnabled` (`trueCLI: --transit-encryption-enabled`) ao criar o cluster de cache usando a operação `CreateCacheCluster` (`CLI: create-cache-cluster`).

Tópicos

- [Visão geral da criptografia em trânsito](#)
- [Condições de criptografia em trânsito](#)
- [Práticas recomendadas de criptografia em trânsito](#)
- [Habilitação da criptografia em trânsito](#)
- [Conectar-se a nós habilitados com criptografia em trânsito usando o Openssl](#)
- [Criação de um cliente TLS Memcached usando Java](#)
- [Criação de um cliente TLS Memcached usando PHP](#)

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.

Condições de criptografia em trânsito

Lembre-se das seguintes restrições de criptografia em trânsito do Amazon ElastiCache quando você planejar sua implementação de cluster autoprojetoado:

- A criptografia em trânsito é compatível com clusters executando as versões 1.6.12 e posteriores do Memcached.
- A criptografia em trânsito é compatível com as versões 1.2 e 1.3 do Transport Layer Security (TLS).
- A criptografia em trânsito é compatível somente em clusters em execução em uma Amazon VPC.
- A criptografia em trânsito é compatível somente em clusters que estejam executando os tipos de nó a seguir.
 - R6g, R5, R4
 - M6g, M5, M4
 - T4g, T3

Para obter mais informações, consulte [Tipos de nó compatíveis](#).

- A criptografia em trânsito é ativada configurando explicitamente o parâmetro `TransitEncryptionEnabled` como `true`.
- Só é possível ativar a criptografia em trânsito em um cluster ao criá-lo. Não é possível ativar e desativar a criptografia em trânsito modificando um cluster.

- Verifique se o cliente de armazenamento em cache dá suporte à conectividade TLS e se você a habilitou na configuração do cliente.

Práticas recomendadas de criptografia em trânsito

- Como a criptografia e descriptografia dos dados requerem processamento nos endpoints, a implementação da criptografia em trânsito pode reduzir o desempenho. Compare seus dados com criptografia em trânsito e sem criptografia para determinar o impacto no desempenho da sua implementação.
- Como criar novas conexões pode ser caro, é possível reduzir o impacto na performance da criptografia em trânsito persistindo suas conexões TLS.

Habilitação da criptografia em trânsito

Para ativar a criptografia em trânsito ao criar um cluster Memcached usando o AWS Management Console, escolha as seguintes opções:

- Escolha o Memcached como seu mecanismo.
- Escolha a versão 1.6.12 ou posterior do mecanismo.
- Em Encryption in transit (Criptografia em trânsito), escolha Enable (Habilitar).

Para o processo passo a passo, consulte [Criação de um cluster do Memcached \(console\)](#).

Conectar-se a nós habilitados com criptografia em trânsito usando o Openssl

Para acessar dados dos nós do ElastiCache para Memcached habilitados com criptografia em trânsito, use clientes que trabalhem com Secure Socket Layer (SSL). Também é possível usar a Openssl s_client no Amazon Linux e no Amazon Linux 2.

Para usar a Openssl s_client para se conectar a um cluster Memcached habilitado com criptografia em trânsito no Amazon Linux 2 ou no Amazon Linux:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Criação de um cliente TLS Memcached usando Java

Para criar um cliente no modo TLS, faça o seguinte para inicializar o cliente com o SSLContext apropriado:

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

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

Criação de um cliente TLS Memcached usando PHP

Para criar um cliente no modo TLS, faça o seguinte para inicializar o cliente com o SSLContext apropriado:

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 * about Auto Discovery and persistent-id.
 */
```



```
/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
 * Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

/* store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
if($tls_client->set('key', 'value', 60)) {
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
}
```

```
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
} else {
    echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
```

Para obter mais informações sobre como usar o cliente PHP, consulte [Instalação do cliente de cluster do ElastiCache para PHP](#).

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](#).

- Disco durante as operações de sincronização e troca

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).

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 só é compatível em caches sem servidor.
- 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).

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](#).

- Se você excluir a chave ou [desabilitá-la](#) e [revogar as concessões](#) da chave que usou para criptografar um cache, o cache ficará irre recuperável. 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.
- 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.

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 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.

Consulte também:

- [Segurança do Amazon VPCs e do ElastiCache](#)
- [Gerenciamento de identidade e acesso no Amazon ElastiCache](#)

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](#).

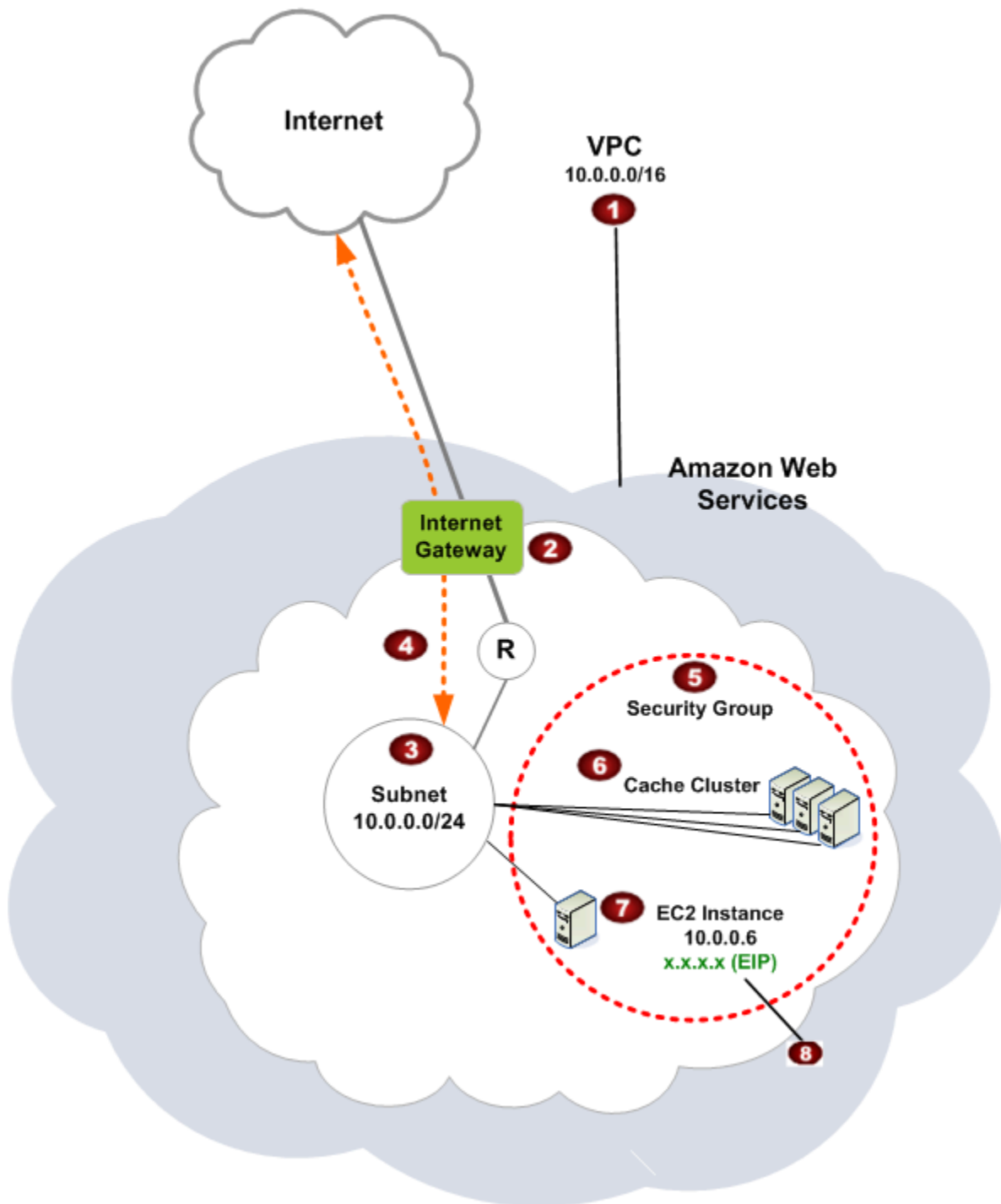
Com a Amazon Virtual Private Cloud, você pode criar uma rede virtual na nuvem da AWS que se assemelha muito com um datacenter tradicional. É possível configurar sua Amazon VPC, incluindo selecionar o intervalo de endereços IP, criar sub-redes e definir tabelas de rotas, gateways de rede e configurações 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:

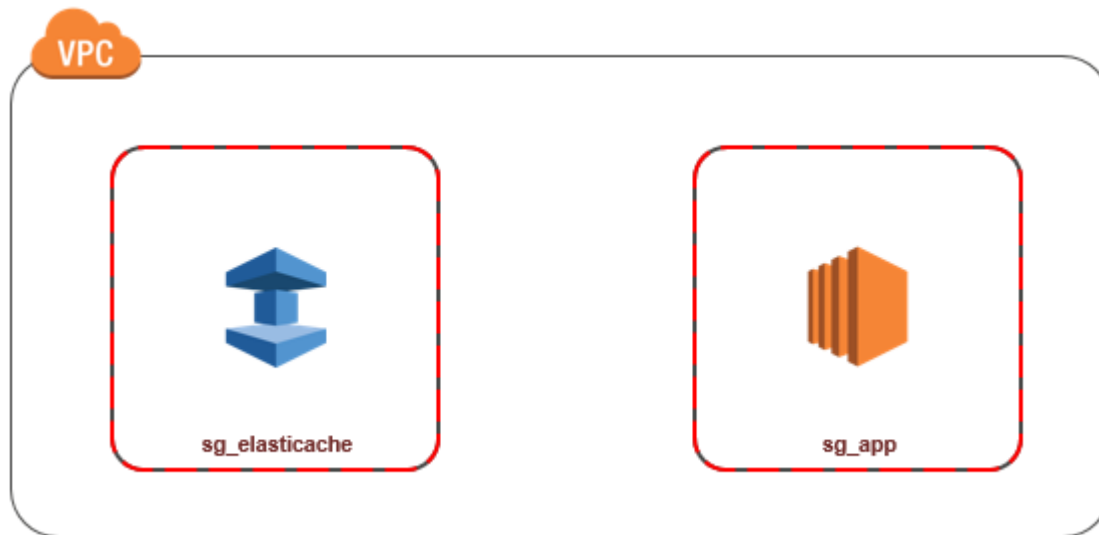
Sumário

- [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 dos caches Memcached é 11211.

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 Memcached é 11211.

- 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.

The screenshot shows the 'Edit inbound rules' dialog in the AWS console. It has a title bar with 'Edit inbound rules' and a close button. Below the title bar are four columns: 'Type' with a dropdown menu showing 'Custom TCP Rule', 'Protocol' with a text input showing 'TCP', 'Port Range' with a text input showing '6379', and 'Source' with a dropdown menu showing 'Custom' and a text input showing 'sg_app'. A dropdown menu is open below the 'Source' field, showing 'sg-99fc5ce6 - sg_app' selected. At the bottom left is an 'Add Rule' button. At the bottom right are 'Cancel' and 'Save' buttons.

Acesso a um cache do ElastiCache quando ele e a instância do Amazon EC2 estão em diferentes Amazon VPCs

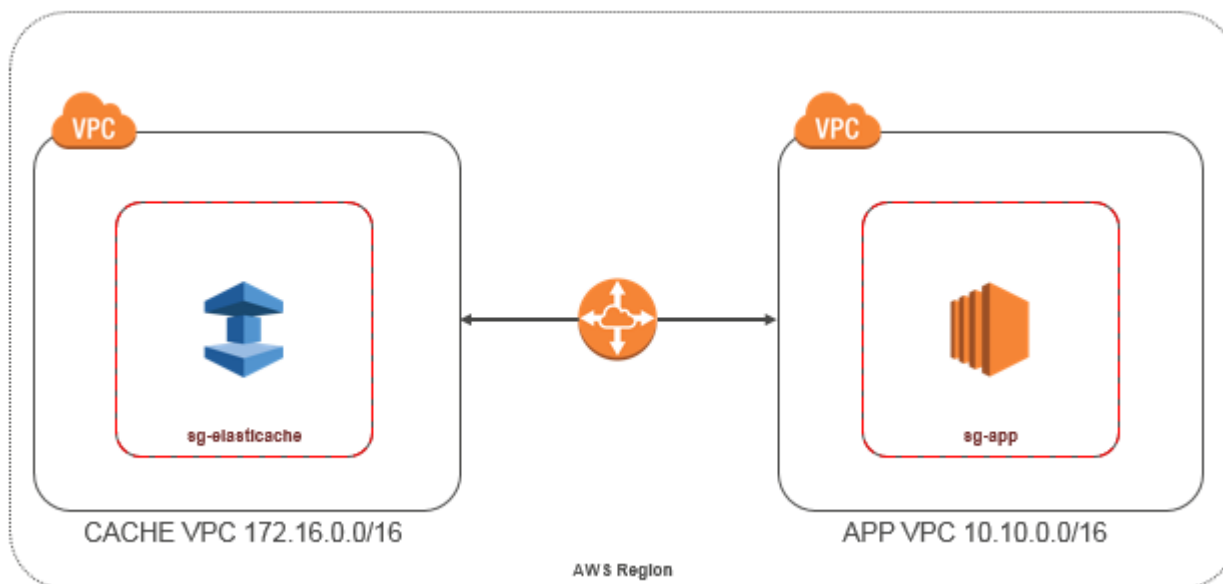
Quando seu cache está em uma VPC diferente da instância do EC2 que você está usando para acessá-la, existem várias maneiras de acessar o cache. Se o cache e a instância do EC2 estiverem em VPCs diferentes, mas na mesma região, você poderá usar o emparelhamento de VPCs. Se o cache e a instância do EC2 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.

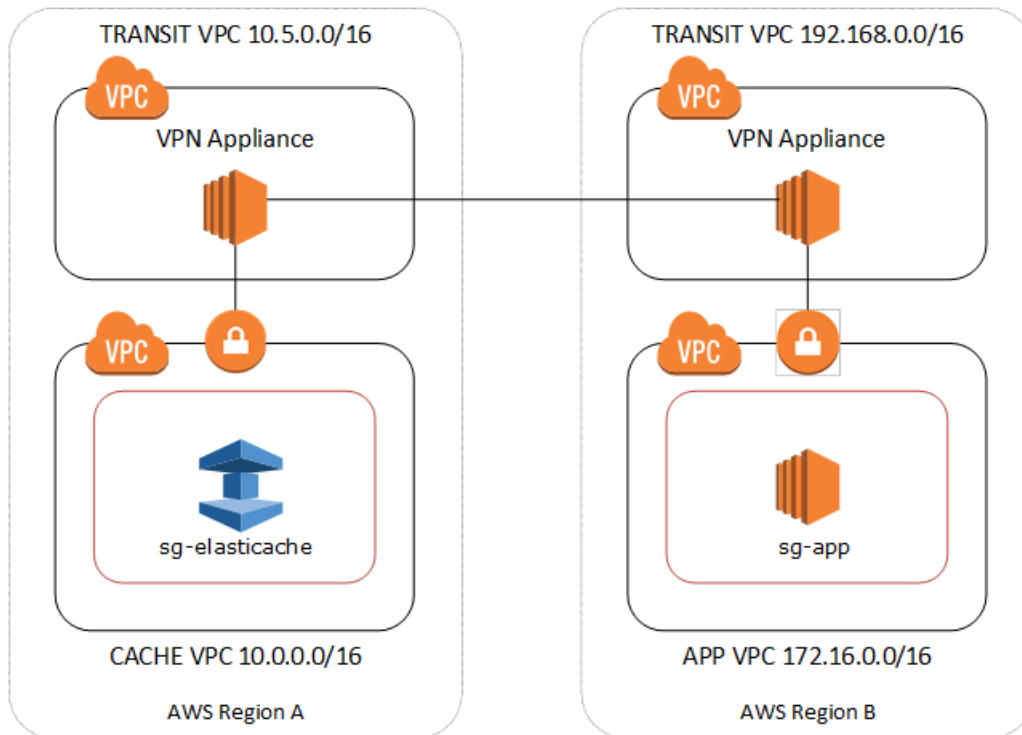
Também é possível criar um anexo da conexão de emparelhamento entre gateways de trânsito em diferentes regiões da AWS. Isso permite que você roteie o tráfego entre os anexos dos gateways de trânsito em regiões diferentes.

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

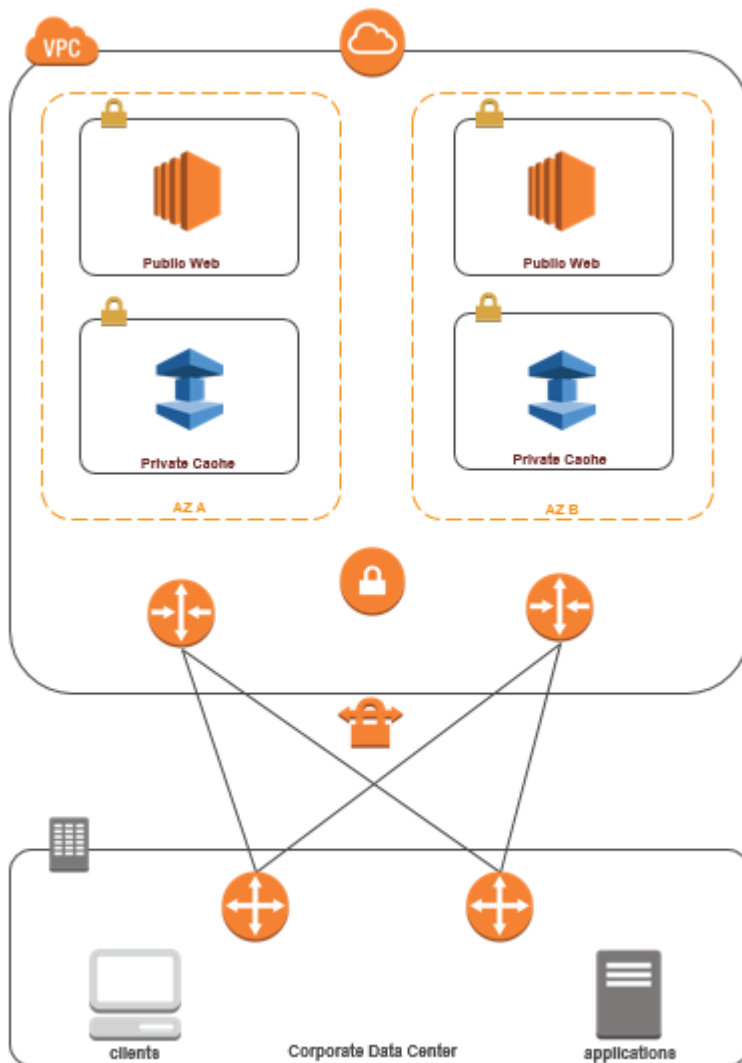
Outro cenário possível é uma arquitetura híbrida em que clientes ou aplicações no datacenter do cliente podem precisar acessar um cache do ElastiCache na VPC. Esse cenário também tem suporte, desde que haja conectividade entre a VPC dos clientes e o datacenter via VPN ou Direct Connect.

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

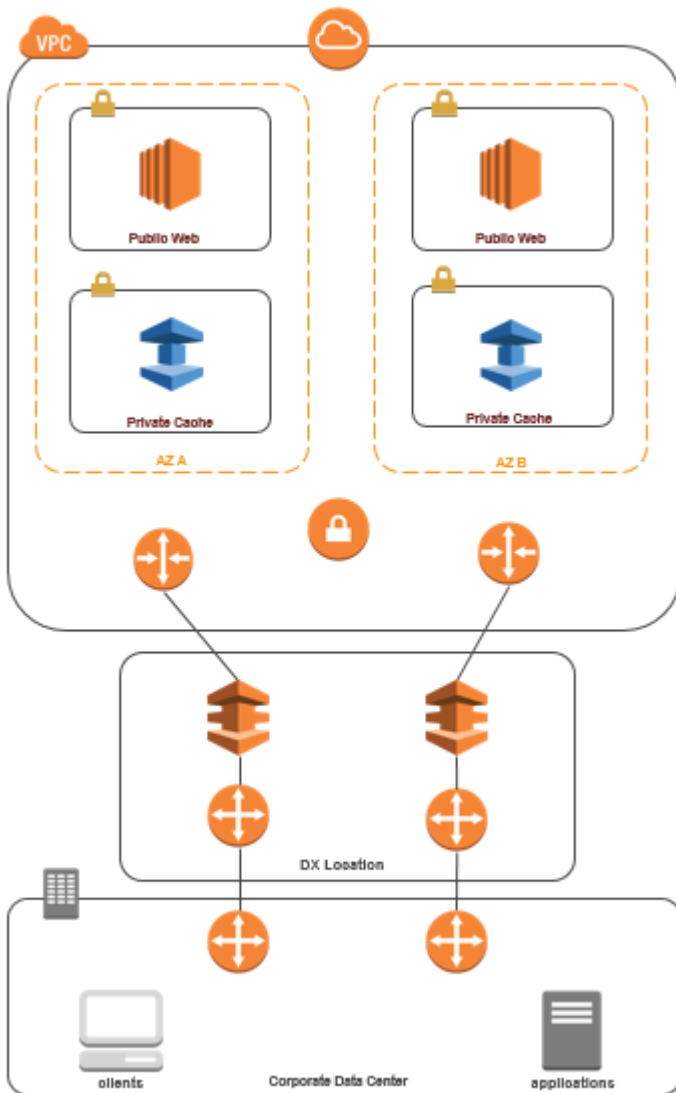
Para acessar um cache em uma VPC a partir do aplicativo no local via conexão VPN

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.

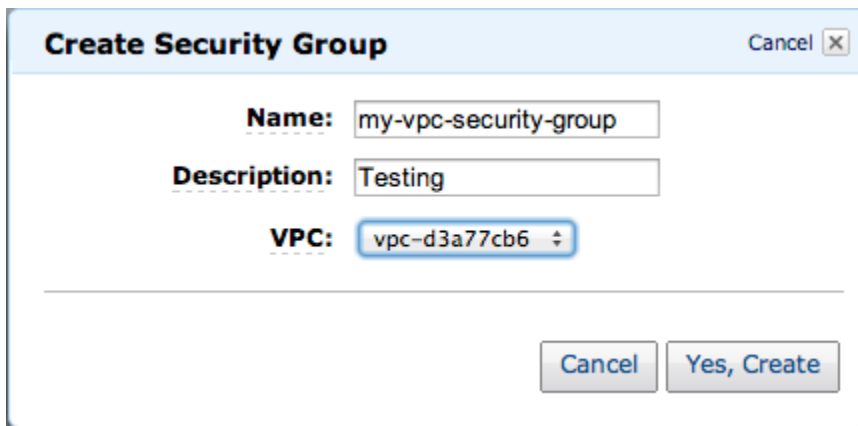
Criação de uma Amazon VPC (console)

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 Cancel

Name: my-vpc-security-group

Description: Testing

VPC: vpc-d3a77cb6

Cancel Yes, Create

- 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 Memcached \(console\)](#).

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 your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description
<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. 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](#) para Memcached.
3. Use o telnet para se conectar ao endpoint do nó de cache pela porta 11211. Substitua o nome do host mostrado abaixo pelo nome do host do nó de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

Agora você está conectado ao mecanismo de cache e pode emitir comandos. Neste exemplo, você adiciona um item de dados ao cache e o obtém logo em seguida. Por último, você se desconectará do nó do cache.

Para armazenar uma chave e um valor, digite as duas linhas a seguir:

```
add mykey 0 3600 28  
This is the value for mykey
```

O mecanismo de cache responde com o seguinte:

```
OK
```

Para recuperar o valor para mykey, digite o seguinte:

```
get mykey
```

O mecanismo de cache responde com o seguinte:

```
VALUE mykey 0 28  
This is the value for my key  
END
```

Para desconectar-se do mecanismo de cache, digite o seguinte:

```
quit
```

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.

Example 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"
    ],
    "Resource": "*"
  }]
}
```

Example 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](#).

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 do Memcached: para ativar um cluster de cache do Memcached, consulte [Criação de um cluster do Memcached \(console\)](#). Na etapa 7.a (Configurações avançadas do Memcached), escolha um grupo de sub-redes da VPC.

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

- [Práticas recomendadas de políticas](#)
- [Uso do console do ElastiCache](#)
- [Permitir que usuários visualizem suas próprias permissões](#)

Práticas recomendadas 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 AWS os clientes da 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.	7 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:DescribeCacheClusters",
      "elasticache:ModifyServerlessCache",
      "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowUserToPassRole",
    "Effect": "Allow",
```

```
    "Action": [ "iam:PassRole" ],
    "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
  }
]
```

A política tem duas instruções:

- A primeira 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 1: permitir a um usuário acesso somente de leitura a recursos do ElastiCache](#)
- [Exemplo 2: permitir que um usuário realize tarefas comuns de administrador do sistema do ElastiCache](#)
- [Exemplo 3: permitir que um usuário acesse todas as ações da API do ElastiCache](#)
- [Exemplo 4: permitir a um usuário chamar a API IAM `CreateServiceLinkedRole`](#)

Exemplo 1: 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 2: 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 3: 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"
        }
      }
    }
  ]
}
```

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 autoprojeto criado será do tipo de nó cache .r5.large.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.r5.large"
          ]
        }
      }
    }
  ]
}
```

```
]
}
```

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"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.t2.micro",
            "cache.t2.medium"
          ]
        }
      }
    }
  ]
}
```

4. `elasticache:EngineVersion`: especifique o uso do mecanismo versão 1.6.6.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineVersion": "1.6.6"
        }
      }
    }
  ]
}
```

5. `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"
      }
    }
  }
]
}

```

6. `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"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    }
  ],
}

```

```

    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheParameterGroupName": "my-org-param-group"
        }
      }
    }
  ]
}

```

7. `elasticache:CreateCacheCluster`: 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"
        ]
      }
    }
  }
]
}

```

8. `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"
      ],
      "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

Para usar o IAM para excluir uma função vinculada a serviço, primeiro confirme se a função não tem nenhum recurso (clusters) associado 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](#)

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.

Exclusão 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 Memcached](#)

- [Recomendações](#)

Mitigar falhas ao executar o Memcached

Ao executar o mecanismo Memcached, você tem as seguintes opções para minimizar o impacto de uma falha. Há dois tipos de falhas que devem ser resolvidas em seus planos de mitigação de falhas: falhas de nó e falhas na zona de disponibilidade.

Mitigar falhas de nós

Os caches sem servidor mitigam automaticamente falhas em nó com uma arquitetura multi-AZ, de maneira que as falhas no nó sejam transparentes para a aplicação. Para mitigar o impacto de uma falha de nó em um cluster autoprotetido, espalhe seus dados em cache por mais nós. Como os clusters autoprotetidos não oferecem suporte para replicação, uma falha de nó sempre resultará em uma certa perda de dados do seu cluster.

Ao criar seu cluster Memcached, você pode criá-lo com 1 a 60 nós, ou mais, mediante solicitação especial. Particionar seus dados em um número maior de nós significa que você perderá menos dados se um nó falhar. Por exemplo, se você particionar seus dados em 10 nós, um único nó armazenará aproximadamente 10% dos seus dados em cache. Nesse caso, uma falha de nó perde aproximadamente 10% do seu cache, que precisa ser substituído quando um nó de substituição é criado e provisionado. Se os mesmos dados tiverem sido armazenados em cache em 3 nós maiores, a falha de um nó perderia aproximadamente 33% dos seus dados em cache.

Se você precisar de mais de 60 nós em um cluster Memcached ou de mais de 300 nós no total em uma AWS região, preencha o formulário de solicitação de aumento de ElastiCache limite em <https://aws.amazon.com/contact-us/elasticache-node-limit-request/>.

Para obter informações sobre como especificar o número de nós em um cluster Memcached, consulte [Criação de um cluster do Memcached \(console\)](#).

Mitigar falhas de zonas de disponibilidade

Os caches sem servidor mitigam automaticamente falhas em zona de disponibilidade com uma arquitetura multi-AZ replicada, de maneira que as falhas em AZ sejam transparentes para a aplicação.

Para mitigar o impacto de uma falha na zona de disponibilidade em um cluster autoprotetido, localize seus nós no maior número possível de zonas de disponibilidade. No caso improvável de uma falha

do AZ, você perderá os dados armazenados em cache nessa AZ, e não os dados armazenados em cache nas outras AZs.

Por que tantos nós?

Se a minha região tem apenas três zonas de disponibilidade, por que preciso de mais de três nós, já que, se uma AZ falhar, perco aproximadamente um terço dos meus dados?

Esta é uma excelente pergunta. Lembre-se de que estamos tentando mitigar dois tipos distintos de falhas: de nó e de zona de disponibilidade. Você está certo. Se os seus dados estiverem espalhados por zonas de disponibilidade e uma delas falhar, você perderá apenas os dados armazenados em cache naquela AZ, independentemente do número de nós que tem. No entanto, se um nó falhar, ter mais nós reduzirá a proporção de dados perdidos.

Não há uma "fórmula mágica" para determinar quantos nós você tem no seu cluster. Você deve ponderar o impacto da perda de dados versus a probabilidade de uma falha versus custo e chegar à sua própria conclusão.

Para obter informações sobre como especificar o número de nós em um cluster Memcached, consulte [Criação de um cluster do Memcached \(console\)](#).

Para obter mais informações sobre regiões e zonas de disponibilidade, consulte [Regiões e zonas de disponibilidade](#).

Recomendações

É recomendável criar caches sem servidor em clusters autoprojeto, pois você obtém automaticamente uma melhor tolerância a falhas sem configuração adicional. No entanto, durante a criação de um cluster autoprojeto, existem dois tipos de falhas para as quais você precisa se planejar: falhas de nós individuais e falhas amplas de zona de disponibilidade. O melhor plano de mitigação de falhas abordará ambos os tipos de falhas.

Minimizar o impacto das falhas em nó

Quando estiver executando o Memcached e particionando seus dados entre nós, quanto mais nós usar, menor será a perda de dados se qualquer um dos nós falhar.

Minimizar o impacto das falhas na zona de disponibilidade

Para minimizar o impacto de uma falha na zona de disponibilidade, nós recomendamos a execução dos nós em quantas zonas de disponibilidade diferentes forem disponíveis. 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.

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 autoprojeto, você configura uma janela de manutenção predefinida para ElastiCache poder aplicar essas atualizações. No entanto, em alguns casos, você pode achar que essa abordagem é muito rígida e que provavelmente restringirá os fluxos de negócios.

Com as atualizações de serviço, você pode controlar quando e quais atualizações são aplicadas aos clusters autoprojeto. Você também pode monitorar o progresso dessas atualizações no ElastiCache cluster selecionado em tempo real.

Gerenciando atualizações de serviços

ElastiCache atualizações de serviço para clusters autoprojeto são lançadas regularmente. Se você tiver um ou mais clusters autoprojeto qualificados para essas atualizações de serviço,

receberá notificações por e-mail, SNS, 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
```

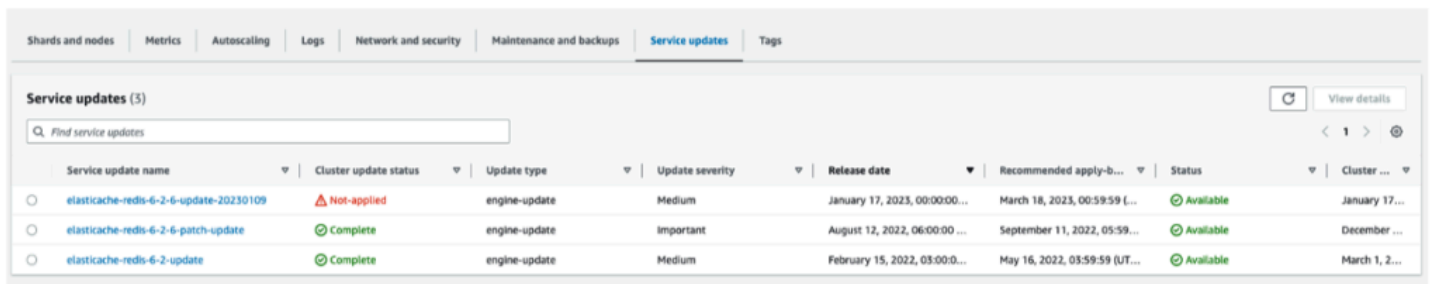
Para obter mais informações, consulte [batch-apply-update-action](#).

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:

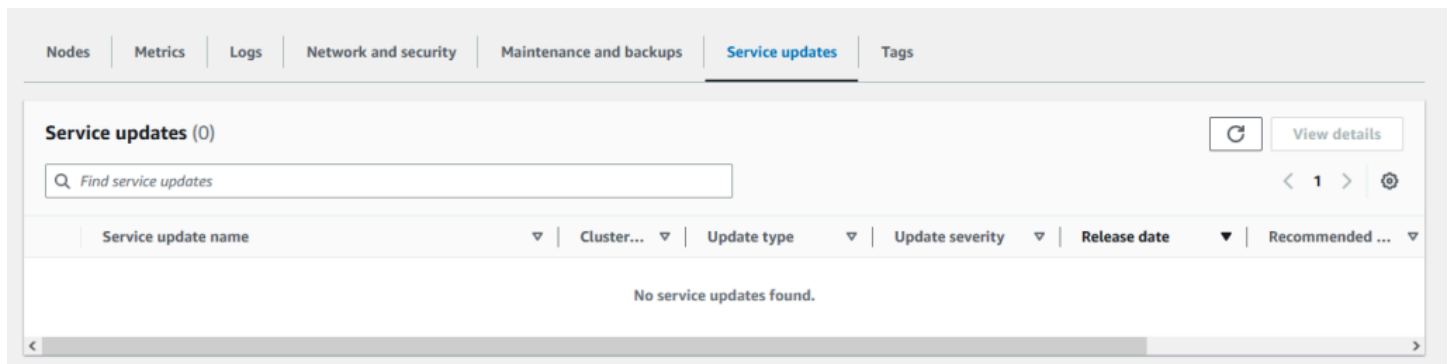
1. Escolha um cluster aplicável na página Redis Clusters
2. Escolha Atualizações de serviço no painel de navegação para ver as atualizações de serviço aplicáveis para esse cluster, se houver.

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-20231019	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 05:59:59 (UT...	Available	March 1, 2...

Se o console exibir “Nenhuma atualização de serviço encontrada”, significa que o cluster ElastiCache for Redis já tem a atualização de serviço mais recente aplicada.



Interromper as atualizações do serviço

É possível interromper as atualizações em clusters, se necessário. Por exemplo, será possível interromper as atualizações se houver uma sobrecarga inesperada nos clusters que estão recebendo atualizações. Ou pode interrompê-las se elas estiverem demorando muito e interrompendo o fluxo de negócios em um horário de pico.

A operação [Stopping \(Interrompendo\)](#) interrompe automaticamente todas as atualizações desses clusters e de quaisquer nós que ainda serão atualizados. Ela continua até a conclusão dos nós que tiverem um status in progress (em andamento). Porém, ela cessa as atualizações em outros nós do mesmo cluster que tenham um status update available (atualização disponível) e as reverte para um status Stopping (Interrompendo).

Quando o fluxo de trabalho Stopping (Interrompendo) for concluído, os nós que tiverem um status Stopping (Interrompendo) serão alterados para um status Stopped (Interrompido). Dependendo do fluxo de trabalho da atualização, alguns clusters não terão nenhum nó atualizado. Outros clusters poderão incluir alguns nós que foram atualizados e outros que ainda terão um status update available (atualização disponível).

Você poderá voltar mais tarde para concluir o processo de atualização conforme seus fluxos de negócios permitirem. Nesse caso, escolha os clusters aplicáveis em que deseja concluir as atualizações e selecione Apply Now (Aplicar agora). Para 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](#).

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](#)
- [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 Memcached.

Métrica	Descrição	Unidade
BytesUsedForCache	O número total de bytes usados pelos dados armazenados no cache.	Bytes

Métrica	Descrição	Unidade
ElastiCacheProcessingUnits	O número total de ElastiCacheProcessingUnits (eCPUs) consumidas pelas solicitações executadas no cache	Contagem
SuccessfulReadRequestLatency	Latência de solicitações de leitura bem-sucedidas.	Microssegundos
SuccessfulWriteRequestLatency	Latência de solicitações de gravação bem-sucedidas	Microssegundos
TotalCmdsCount	Contagem total de todos os comandos executados no cache	Contagem
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
NewConnections	O número total de conexões que foram aceitas pelo servidor durante esse período.	Contagem
CurrItems	O número de itens no cache.	Contagem
NetworkBytesIn	Total de bytes transferidos para dentro do cache	Bytes
NetworkBytesOut	Total de bytes transferidos para fora do cache	Bytes

Métrica	Descrição	Unidade
Evictions	A contagem de chaves removidas pelo cache	Contagem
Reclaimed	O número de chaves expiradas pelo cache	Contagem

Métricas no nível de comando

O ElastiCache também emite as métricas a seguir no nível de comando do Memcached

Métrica	Descrição	Unidade
CmdGet	O número de comandos get recebidos pelo cache.	Contagem
CmdSet	O número de comandos set recebidos pelo cache.	Contagem
CmdTouch	O número de comandos touch recebidos pelo cache.	Contagem
GetHits	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
GetMisses	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
IncrHits	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem

Métrica	Descrição	Unidade
IncrMisses	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
DecrHits	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
DecrMisses	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
DeleteHits	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
DeleteMisses	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
TouchHits	O número de chaves que foram tocadas e receberam um novo horário de expiração.	Contagem
TouchMisses	O número de chaves que foram tocadas e não foram encontradas.	Contagem

Métrica	Descrição	Unidade
CasHits	O número de solicitações cas recebidas pelo cache em que a chave solicitada foi encontrada e o valor de cas correspondia.	Contagem
CasMisses	O número de solicitações cas recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
CasBadval	O número de solicitações cas recebidas pelo cache em que o valor de cas não correspondeu ao valor de cas armazenado.	Contagem
CmdFlush	O número de comandos flush recebidos pelo cache.	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 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 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.

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.

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>.
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>. 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.
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

Detail-Type	Lista de recursos	Categoria	Origem	Message
				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>.
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.

Métricas e eventos de clusters autoprotetados

Esta seção descreve as métricas, os eventos e os logs que você pode ver ao trabalhar com clusters autoprotetados.

Tópicos

- [Métricas para clusters autoprotetados](#)
- [Eventos para clusters autoprotetados](#)
- [Monitoramento de uso com métricas do CloudWatch](#)
- [Monitoramento de eventos do ElastiCache pelo Amazon SNS](#)

Métricas para clusters autoprotetados

Quando você projeta clusters autoprotetados, 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 para Memcached, consulte [Métricas em nível de host](#).

Para obter mais informações sobre métricas do Memcached no nível de host, consulte [Métricas para o Memcached](#).

Eventos para clusters autoprotetados

O ElastiCache registra eventos de logs relacionados aos caches autoprotetados. Ao trabalhar com clusters autoprotetados, 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 autoprotetados não são publicados no Amazon EventBridge.

As informações do evento de cluster autoprotetado 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 `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.

Nome do evento	Message	Descrição
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.
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 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: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 medidas você pode tomar, consulte Replacing cache nodes for Memcached .
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 medidas você pode tomar, consulte Replacing cache nodes for Memcached .
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.

Monitoramento de 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 o Memcached](#)
- [Quais métricas devo monitorar?](#)
- [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.

Consulte também

- [Métricas para o Memcached](#)

Métrica	Descrição	Unidade
CPUUtilization	A percentagem de utilização da CPU para o host inteiro.	Percentual
CPUCreditBalance	O número de créditos ganhos de CPU que uma instância acumulou desde que foi executada ou iniciada. Para a T2 Padrão, o CPUCredit Balance também inclui o número de créditos de execução que foram acumulados. Os créditos são acumulados no saldo de créditos após terem sido ganhos e são removidos do saldo de créditos quando são	Créditos (minutos de vCPU)

Métrica	Descrição	Unidade
	<p>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 CPUCreditBalance 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>	
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>	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

Métrica	Descrição	Unidade
NetworkBytesIn	O número de bytes que o host leu da rede.	Bytes
NetworkBytesOut	O número de bytes enviados em todas as interfaces de rede pela instância.	Bytes
NetworkPacketsIn	O número de pacotes recebidos em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de entrada em termos do número de pacotes em uma única instância.	Contagem
NetworkPacketsOut	O número de pacotes enviados em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de saída em termos do número de pacotes em uma única instância.	Contagem
NetworkBandwidthIn AllowanceExceeded	Número de pacotes moldados porque a largura de banda agregada de entrada excedeu o máximo para a instância.	Contagem
NetworkConntrackAllowanceExceeded	Número de pacotes moldados 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
NetworkBandwidthOut AllowanceExceeded	Número de pacotes moldados porque a largura de banda agregada de saída excedeu o máximo para a instância.	Contagem
Network Packets Per Second Allowance Exceeded	Número de pacotes moldados porque o valor bidirecional de pacotes por segundo excedeu o máximo para a instância.	Contagem

Métrica	Descrição	Unidade
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
SwapUsage	A quantidade de troca usada no host.	Bytes

Métricas para o Memcached

O namespace AWS/ElastiCache inclui as métricas do Memcached a seguir.

O ElastiCache namespace AWS/inclui as seguintes métricas derivadas do comando de estatísticas do Memcached. Cada métrica é calculada no nível de nó de cache.

Consulte também

- [Métricas em nível de host](#)

Métrica	Descrição	Unidade
BytesReadIntoMemcached	O número de bytes que foram lidos da rede pelo nó de cache.	Bytes
BytesUsedForCacheItems	O número de bytes usados para armazenar itens de cache.	Bytes
BytesWrittenOutFromMemcached	O número de bytes que foram gravados na rede pelo nó de cache.	Bytes

Métrica	Descrição	Unidade
CasBadval	O número de solicitações de CAS (verificar e definir) recebidas pelo cache em que o valor de Cas não correspondeu ao valor de Cas armazenado.	Contagem
CasHits	O número de solicitações de Cas recebidas pelo cache em que a chave solicitada foi encontrada e o valor de Cas correspondia.	Contagem
CasMisses	O número de solicitações Cas recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
CmdFlush	O número de comandos flush recebidos pelo cache.	Contagem
CmdGet	O número de comandos get recebidos pelo cache.	Contagem
CmdSet	O número de comandos set recebidos pelo cache.	Contagem
CurrConnections	<p>A contagem do número de conexões do cache em determinado momento. ElastiCache usa duas a três conexões para monitorar o cluster.</p> <p>Além do mencionado acima, o Memcached cria um número de conexões internas iguais ao dobro dos threads usados para o tipo de nó. A contagem de threads para os vários tipos de nó pode ser vista no <code>Nodetype Specific Parameters</code> do grupo de parâmetros aplicável.</p> <p>O total de conexões é a soma das conexões de cliente, as conexões para monitoramento e as conexões internas mencionadas acima.</p>	Contagem

Métrica	Descrição	Unidade
CurrItems	A contagem do número de itens atualmente armazenados no cache.	Contagem
DecrHits	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
DecrMisses	O número de solicitações de decremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
DeleteHits	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
DeleteMisses	O número de solicitações de exclusão recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
Evictions	O número de itens não expirados removidos pelo cache para permitir espaço para novas gravações.	Contagem
GetHits	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem
GetMisses	O número de solicitações de obtenção recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
IncrHits	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada foi encontrada.	Contagem

Métrica	Descrição	Unidade
IncrMisses	O número de solicitações de incremento recebidas pelo cache em que a chave solicitada não foi encontrada.	Contagem
Reclaimed	O número de itens expirados removidos pelo cache para permitir espaço para novas gravações.	Contagem

Para o Memcached 1.4.14, as seguintes métricas adicionais são fornecidas.

Métrica	Descrição	Unidade
BytesUsedForHash	O número de bytes atualmente usados por tabelas de hash.	Bytes
CmdConfigGet	O número cumulativo de solicitações config get.	Contagem
CmdConfigSet	O número cumulativo de solicitações config set.	Contagem
CmdTouch	O número cumulativo de solicitações touch.	Contagem
CurrConfig	O número atual de configurações armazenadas.	Contagem
EvictedUnfetched	O número de itens válidos removidos do cache menos usado recentemente (LRU) que nunca foram tocados após serem definidos.	Contagem
ExpiredUnfetched	O número de itens expirados recuperados do LRU que nunca foram tocados após serem definidos.	Contagem
SlabsMoved	O número total de páginas de prancha que foram movidas.	Contagem

Métrica	Descrição	Unidade
TouchHits	O número de chaves que foram tocadas e receberam um novo horário de expiração.	Contagem
TouchMisses	O número de itens que foram tocados, mas não foram encontrados.	Contagem

O ElastiCache namespace AWS/inclui as seguintes métricas calculadas em nível de cache.

Métrica	Descrição	Unidade
NewConnections	O número de novas conexões recebidas pelo cache. Isso é derivado da estatística <code>total_connections</code> do Memcached registrando a alteração em <code>total_connections</code> durante um período. Isso sempre será pelo menos 1, devido a uma conexão reservada para ElastiCache a.	Contagem
NewItems	O número de novos itens armazenados pelo cache. Isso é derivado da estatística <code>total_items</code> do Memcached registrando a alteração em <code>total_items</code> ao longo de um período.	Contagem
UnusedMemory	<p>A quantidade de memória não usada pelos dados. Isso é derivado das estatísticas <code>limit_maxbytes</code> e <code>bytes</code> do Memcached subtraindo <code>bytes</code> de <code>limit_maxbytes</code>.</p> <p>Como a sobrecarga do Memcached usa memória além da usada pelos dados, não UnusedMemory deve ser considerada a quantidade de memória disponível para dados adicionais. Você pode passar por remoções, mesmo que você ainda tenha alguma memória não utilizada.</p>	Bytes

Métrica	Descrição	Unidade
	Para obter informações mais detalhadas, consulte Memcached item memory usage .	

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](#)
- [SwapUsage](#)
- [Evictions](#)
- [CurrConnections](#)

CPUUtilization

Essa é uma métrica em nível de host relatada como uma porcentagem. Para obter mais informações, consulte [Métricas em nível de host](#).

Como o Memcached é multi-threaded, essa métrica pode atingir até 90%. Se você exceder esse limite, dimensione seu cluster de cache usando um tipo de nó de cache maior ou expanda adicionando mais nós de cache.

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 ela exceder esse limite, recomendamos que você aumente o valor do parâmetro `ConnectionOverhead`.

Evictions

Esta é uma métrica do mecanismo de cache. Recomendamos que você determine seu próprio limite de alarme para essa métrica com base nas necessidades do seu aplicativo.

se você exceder o limite escolhido, dimensione seu cluster usando um tipo de nó maior ou expanda adicionando mais nós.

CurrConnections

Esta é uma métrica do mecanismo de cache. Recomendamos que você determine seu próprio limite de alarme para essa métrica com base nas necessidades do seu aplicativo.

Um número crescente de CurrConnections pode indicar um problema em seu aplicativo; você precisará investigar o comportamento do aplicativo para resolver esse problema.

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.

 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.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=false  
&CacheClusterId=my-cluster  
&NotificationTopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

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

Habilitação e desabilitação de notificações do Amazon SNS

Você pode ativar ou desativar notificações para um cluster. Os procedimentos a seguir mostram como desativar notificações do Amazon SNS.

Habilitação e desabilitação de notificações do Amazon SNS (console)

Para 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 Memcached, no painel de navegação, escolha Memcached.
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 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 medidas você pode tomar, consulte Replacing cache nodes for Memcached .
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 medidas você pode tomar, consulte Replacing cache nodes for Memcached .
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"
  },

```

```
"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
Nós por região	300
Nós por cluster	40
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

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://elasticache.amazonaws.com>). Uma `MissingAuthenticationTokenException` ou um Erro do servidor interno 500 confirma que o serviço está disponível e respondendo a solicitações.

- Verificação da estrutura de sua solicitação.

Cada operação do ElastiCache tem uma página de referência na Referência da API do ElastiCache. Verifique novamente se você está usando os parâmetros corretamente. Para conceder ideias sobre o que pode estar errado, consulte as amostras de solicitações ou cenários de usuários para ver se esses exemplos estão realizando operações similares.

- Verificação do fórum.

O ElastiCache tem um fórum de discussão onde você pode procurar soluções para os problemas que outros usuários enfrentaram ao longo do caminho. Para exibir o fórum, consulte

<https://forums.aws.amazon.com/> .

Configuração da interface de linha de comando do ElastiCache

Esta seção descreve os pré-requisitos para executar as ferramentas da linha de comando, em que lugar obtê-las e como configurar as ferramentas e o ambiente, e inclui ainda uma série de exemplos comuns de uso das ferramentas.

Siga as instruções neste tópico somente se for usar a AWS CLI para o ElastiCache.

Important

A CLI (Command Line Interface) do Amazon ElastiCache não oferece suporte a melhorias do ElastiCache após a versão 2014-09-30 da API. Para usar a funcionalidade do ElastiCache mais recente na linha de comando, use a opção [Interface de linha de comando da AWS](#).

Tópicos

- [Pré-requisitos](#)
- [Obtenção das ferramentas da linha de comando](#)
- [Configuração das ferramentas](#)
- [Como fornecer credenciais para as ferramentas](#)
- [Variáveis de ambiente](#)

Pré-requisitos

Este documento pressupõe que você pode trabalhar em um ambiente Linux/UNIX ou Windows. As ferramentas da linha de comando do Amazon ElastiCache também funcionam no Mac OS X, que é um ambiente baseado no UNIX; contudo, não há nenhuma instrução específica ao Mac OS X neste guia.

Como convenção, todos os textos da linha de comando recebem prefixo com um prompt da linha de comando genérico **PROMPT>** . O prompt da linha de comando real da sua máquina pode ser diferente. Também usamos **\$** para indicar um comando específico do Linux/UNIX e **C:\>** para um comando específico do Windows. O exemplo de saída resultante do comando é mostrado imediatamente após, sem qualquer prefixo.

O Java Runtime Environment

As ferramentas da linha de comando usadas neste guia requerem a execução do Java versão 5 ou posterior. Uma instalação do JRE ou do JDK é aceitável. Para ver e fazer download de JREs para uma série de plataformas, incluindo Linux/UNIX e Windows, consulte [Java SE Downloads](#).

Configuração da variável home do Java

As ferramentas da linha de comando dependem de uma variável de ambiente (JAVA_HOME) para localizar o Java Runtime. Essa variável de ambiente deve ser definida para o caminho completo do diretório que contém um subdiretório denominado bin, que, por sua vez, contém o executável java (no Linux e no UNIX) ou o executável java.exe (no Windows).

Para configurar da variável Java Home

1. Configure a variável Java Home.

- No Linux e UNIX, digite o comando a seguir:

```
$ export JAVA_HOME=<PATH>
```

- No Windows, digite o seguinte comando:

```
C:\> set JAVA_HOME=<PATH>
```

2. Confirme a configuração do caminho executando \$JAVA_HOME/bin/java -version e verificando a saída.

- No Linux/UNIX, você verá uma saída semelhante à seguinte:

```
$ $JAVA_HOME/bin/java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

- No Windows, você verá uma saída semelhante à seguinte:

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

Obtenção das ferramentas da linha de comando

As ferramentas da linha de comando estão disponíveis como arquivo ZIP no [site Ferramentas de desenvolvedor do ElastiCache](#). Essas ferramentas são escritas em Java e incluem scripts de shell para Windows 2000/XP/Vista/Windows 7, Linux/UNIX e Mac OSX. O arquivo ZIP é independente e nenhuma instalação é necessária; basta fazer download do arquivo ZIP e decompactá-lo em um diretório de seu computador local.

Configuração das ferramentas

As ferramentas da linha de comando dependem de uma variável de ambiente (AWS_ELASTICACHE_HOME) para localizar bibliotecas compatíveis. Você precisa definir essa variável de ambiente para poder usar as ferramentas. Indique o caminho do diretório no qual você descompactou as ferramentas da linha de comando. Esse diretório é chamado ElastiCacheCli-A.B.nnnn (A, B e n são os números de versão/release) e contém os subdiretórios denominados bin e lib.

Para definir a variável de ambiente AWS_ELASTICACHE_HOME

- Abra uma janela de linha de comando e insira um dos seguintes comandos para definir a variável de ambiente AWS_ELASTICACHE_HOME.
 - No Linux e UNIX, digite o comando a seguir:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- No Windows, digite o seguinte comando:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Para facilitar o uso das ferramentas, é recomendável adicionar o diretório BIN das ferramentas ao CAMINHO do sistema. O restante deste guia pressupõe que o diretório BIN encontra-se no caminho do sistema.

Para adicionar o diretório BIN das ferramentas ao caminho do sistema

- Insira os comandos a seguir para adicionar o diretório BIN das ferramentas ao CAMINHO do sistema.

- No Linux e UNIX, digite o comando a seguir:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- No Windows, digite o seguinte comando:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

Note

As variáveis do ambiente Windows são redefinidas quando você fecha a janela de comando. Talvez você queira defini-las permanentemente. Consulte a documentação da versão de seu Windows para obter mais informações.

Note

Os caminhos que contêm um espaço devem ser colocados entre aspas duplas; por exemplo: “C:\Program Files\Java”

Como fornecer credenciais para as ferramentas

As ferramentas da linha de comando precisam da chave de acesso e da chave de acesso secreta da AWS fornecidas com sua conta da AWS. Você pode obtê-las usando a linha de comando ou um arquivo de credenciais localizado no sistema local.

A implantação inclui um arquivo de modelo `${AWS_ELASTICACHE_HOME}/credential-file-path.template` que você precisa alterar com suas informações. A seguir encontra-se o conteúdo do arquivo de modelo:

```
AWSAccessKeyId=<Write your AWS access ID>  
AWSSecretKey=<Write your AWS secret key>
```

⚠ Important

No UNIX, limite as permissões ao proprietário do arquivo de credenciais:

```
$ chmod 600 <the file created above>
```

Com o arquivo de credenciais configurado, você precisará definir a variável de ambiente `AWS_CREDENTIAL_FILE` para que as ferramentas do ElastiCache possam encontrar suas informações.

Para definir a variável de ambiente `AWS_CREDENTIAL_FILE`

1. Defina a variável de ambiente:

- No Linux e UNIX, atualize a variável usando o seguinte comando:

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- No Windows, defina a variável usando o seguinte comando:

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. Verifique se sua configuração está funcionando corretamente com o seguinte comando:

```
elasticache --help
```

Você deve examinar a página de uso de todos os comandos do ElastiCache.

Variáveis de ambiente

As variáveis de ambiente podem ser úteis para scripting, configuração de padrões ou para substituí-los temporariamente.

Além da variável de ambiente `AWS_CREDENTIAL_FILE`, a maioria das ferramentas de API incluídas com a interface da linha de comando do ElastiCache comporta as seguintes variáveis:

- `EC2_REGION`: a região da AWS a ser usada.

- `AWS_ELASTICACHE_URL`: o URL a ser usado para a chamada de serviço. Não é necessária para especificar um endpoint regional diferente quando `EC2_REGION` for especificada ou quando se passa o parâmetro `--region`.

Os exemplos a seguir mostram como se define a variável de ambiente `EC2_REGION` para configurar a região usada pelas ferramentas de API:

Linux, OSX ou UNIX

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

Mensagens 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 Memcached. 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](#)
- [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: 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 Memcached](#).

Notificações

Este tópico discute as notificações do ElastiCache nas quais você possa estar interessado. Uma notificação é uma situação ou evento que, na maioria dos casos, é temporário, durando apenas até que uma solução seja encontrada e implementada. Em geral, as notificações têm uma data de início e uma data de resolução, após a qual elas deixam de ser relevantes. Qualquer notificação pode ou não ser relevante para você. Recomendamos uma diretriz de implementação que, se seguida, vai melhorar o desempenho do seu cluster.

Notificações não anunciam recursos ou funcionalidades novas ou melhoradas do ElastiCache.

Notificações gerais do ElastiCache

Atualmente não há nenhuma notificação importante do ElastiCache que não seja específica ao mecanismo.


Notificações do ElastiCache for Memcached

As notificações ElastiCache a seguir são específicas ao mecanismo Memcached.

Notificações específicas do ElastiCache for Memcached

- [Alerta: crawler de LRU do Memcached causando falhas de segmentação](#)

Alerta: crawler de LRU do Memcached causando falhas de segmentação

 Data do alerta: 28 de fevereiro de 2017

Em algumas circunstâncias, seu cluster pode exibir instabilidade com uma falha de segmentação no Crawler de LRU do Memcached. Este é um problema no mecanismo Memcached que existe há algum tempo. O problema ficou aparente no Memcached 1.4.33, quando o Crawler de LRU foi habilitado por padrão.

Se você estiver enfrentando esse problema, recomendamos desabilitar o Crawler de LRU até que haja uma correção. Para fazer isso, use `lru_crawler disable` na linha de comando ou modifique o valor do parâmetro `lru_crawler` (preferencial).

Data de resolução:

Resolução:

- Versão da API: 02/02/2015
- Última atualização da documentação: 27 de novembro de 2023

A tabela a seguir descreve mudanças importantes em cada versão do Guia do ElastiCache Usuário do Memcached ElastiCache . Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em o feed RSS.

Atualizações recentes ElastiCache do Memcached

Alteração	Descrição	Data
Support ElastiCache para Memcached 1.6.22	ElastiCache para Memcached agora oferece suporte ao Memcached 1.6.22. Ele inclui correções de erros cumulativas da versão 1.6.18 . Para obter mais informações, consulte Memcached versão 1.6.22 .	10 de janeiro de 2024
ElastiCache para Memcached agora suporta a criação de caches sem servidor	Agora você pode criar caches sem servidor, que simplificam o gerenciamento de cache e escalam instantaneamente para dar suporte a aplicações mais exigentes . Para obter mais informações, consulte Choosing between deployment options . Como parte desse recurso, novas permissões foram adicionadas a ElastiCacheServiceRolePolicy e a AmazonElastiCacheFullAccess para permitir a associação de caches	27 de novembro de 2023

sem servidor com endpoints da VPC gerenciados. Além disso, as permissões foram adicionadas para dar suporte a uma experiência revisada do console usando a política AmazonElastiCacheFullAccess .

[Support ElastiCache para Memcached 1.6.17](#)

ElastiCache para Memcached agora oferece suporte ao Memcached 1.6.17. Ele inclui correções de erros cumulativas do [Memcached versão 1.6.12](#). Para obter mais informações, consulte [Memcached versão 1.6.17](#).

18 de janeiro de 2023

[ElastiCache para Memcached 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 Memcached versão 1.6.6 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 para Memcached
agora oferece suporte à
criptografia em trânsito](#)

A criptografia em trânsito é um recurso opcional que permite aumentar a segurança de seus dados nos pontos mais vulneráveis quando eles estiverem em trânsito de um local para outro. Ela é compatível com as versões 1.6.12 e posteriores do Memcached. Para obter mais informações, consulte [Criptografia ElastiCache em trânsito \(TLS\)](#).

26 de maio de 2022

[ElastiCache 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

[Suporte à marcação de recursos e chaves de condição](#)

ElastiCache agora oferece suporte à marcação para ajudar você a gerenciar seus clusters e outros ElastiCache recursos. Para obter mais informações, consulte [Como marcar seus ElastiCache recursos](#). ElastiCache também introduz suporte para chaves de condição. Você pode especificar as condições que determinam como uma política do IAM entra em vigor. Para obter mais informações, consulte [Uso de chaves de condição](#).

7 de abril de 2021

[Suporte ao Memcached 1.6.6.](#)

ElastiCache para Memcached agora oferece suporte ao Memcached 1.6.6. Ele inclui correções de erros cumulativas das versões Memcached 1.5.16. Para obter mais informações, consulte [Memcached versão 1.6.6](#).

12 de novembro de 2020

[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 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 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 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 usa sua janela de manutenção para programar a atualização automática dos clusters aplicáveis. Para obter mais informações, consulte [Atualizações de autoatendimento](#).

13 de maio de 2020

[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

[ElastiCache para Memcached 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 de serviços, principalmente durante o pico dos fluxos de negócios, e ajudará a garantir que você mantenha a conformidade com as atualizações de segurança. Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache](#).

9 de outubro de 2019

[Support ElastiCache para Memcached 1.5.16](#)

ElastiCache para Memcached agora oferece suporte ao Memcached 1.5.16. Inclui correções de erros cumulativas das versões [Memcached 1.5.14](#) e [Memcached 1.5.15](#). Para obter mais informações, consulte [Memcached na versão 1.5.16](#).

6 de setembro 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

[Support ElastiCache para Memcached 1.5.10](#)

ElastiCache para Memcached agora oferece suporte ao Memcached 1.5.10. Ele inclui suporte para os parâmetros `no_modern` e `inline_ascii_resp`. Para obter mais informações, consulte [Memcached na versão 1.5.10](#).

14 de novembro 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

A tabela a seguir descreve as mudanças importantes no Guia do ElastiCache usuário do Memcached ElastiCache .

Alteração	Descrição	Alterado em
Suporte para a região Ásia-Pacífico (Osaka-local).	ElastiCache adicionou suporte para a região Ásia-Pacífico (local de Osaka). Atualmente, a região Ásia-Pacífico (Osaka) oferece suporte a uma única zona de disponibilidade e somente por meio de convite. Para obter mais informações, consulte: <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).	ElastiCache adicionou suporte para a região da UE (Paris). Para obter mais informações, consulte: <ul style="list-style-type: none"> • Supported Regions (Regiões compatíveis) 	18 de dezembro de 2017

Alteração	Descrição	Alterado em
	<ul style="list-style-type: none">• Tipos de nó de cache compatíveis	
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">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	11 de dezembro de 2017
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 do tipo de nó do Memcached	20 de novembro de 2017

Alteração	Descrição	Alterado em
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>	24 de abril de 2017
Suporte para Memcached 1.4.34	<p>ElastiCache adiciona suporte ao Memcached versão 1.4.34, que incorpora várias correções às versões anteriores do Memcached.</p> <p>Para obter mais informações, consulte as notas de lançamento do Memcached 1.4.34 em Memcached on. GitHub</p>	10 de abril de 2017
Suporte para Memcached 1.4.33	<p>ElastiCache adiciona suporte para a versão 1.4.33 do Memcached. Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Memcached versão 1.4.33• Parâmetros adicionados do Memcached 1.4.33	20 de dezembro de 2016
Suporte para a região oeste da Europa (Londres)	<p>ElastiCache adiciona suporte para a região da UE (Londres). No momento, há suporte apenas para os tipos de nós T2 e M4. Para obter mais informações, consulte:</p> <ul style="list-style-type: none">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	13 de dezembro de 2016

Alteração	Descrição	Alterado em
Suporte à região Canadá (Montreal)	ElastiCache adiciona suporte para a região do Canadá (Montreal). Somente os tipos de nó M4 e T2 são atualmente suportados nesta AWS região. Para obter mais informações, consulte: <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	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: <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)	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: <ul style="list-style-type: none">• Supported Regions (Regiões compatíveis)• Tipos de nó de cache compatíveis	17 de outubro de 2016
Suporte para tipo de nós M4	ElastiCache adiciona suporte para a família M4 de tipos de nós na maioria das AWS regiões suportadas peloElastiCache. Você pode adquirir os tipos de nó M4 sob demanda ou como nós de cache reservados. Para obter mais informações, consulte: <ul style="list-style-type: none">• Tipos de nó de cache compatíveis• Parâmetros específicos do tipo de nó do Memcached	3 de agosto de 2016

Alteração	Descrição	Alterado em
Suporte para a região de Mumbai	ElastiCache adiciona suporte para a região Ásia-Pacífico (Mumbai). Para obter mais informações, consulte: <ul style="list-style-type: none">• Tipos de nó de cache compatíveis• Parâmetros específicos do tipo de nó do Memcached	27 de junho de 2016
Suporte para tipos de nós R3	ElastiCache adiciona suporte para tipos de nós R3 na região da China (Pequim) e na região da América do Sul (São Paulo). Para obter mais informações, consulte Supported cache node types .	16 de março de 2016
Acessando ElastiCache usando uma função Lambda	Foi adicionado um tutorial sobre como configurar uma função Lambda para ElastiCache acessar em uma Amazon VPC. Para ter mais informações, consulte Tutoriais e vídeos do ElastiCache .	12 de fevereiro 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
Suporte para o Memcached 1.4.28.	ElastiCache adiciona suporte para o Memcached versão 1.4.24 e melhorias no Memcached desde a versão 1.4.14. Esta versão adiciona suporte para gerenciamento de cache LRU (menos utilizado recentemente) como uma tarefa em segundo plano, a opção por jenkins ou murmur3 como algoritmo de hashing, novos comandos e correções de diversos erros. Para obter mais informações, as Notas de release do Memcached .	27 de agosto de 2015

Alteração	Descrição	Alterado em
Suporte para a Descoberta automática do Memcached usando PHP 5.6	Esta versão da Amazon ElastiCache adiciona suporte ao cliente Memcached Auto Discovery para PHP versão 5.6. Para ter mais informações, consulte Compilação do código-fonte para o cliente de cluster do ElastiCache para PHP .	29 de julho de 2015
Novo tópico: Acessando ElastiCache de fora AWS	Foi adicionado um novo tópico sobre como acessar ElastiCache recursos de fora AWS. Para obter mais informações, consulte Acessando ElastiCache 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: 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
Suporte para tags de alocação de custos	ElastiCache adiciona suporte para tags de alocação de custos. Para ter mais informações, consulte Monitoramento de custos com tags de alocação de custos .	9 de fevereiro de 2015
Support para a região AWS GovCloud (Oeste dos EUA)	ElastiCache adiciona suporte para a região AWS GovCloud (Oeste dos EUA) (us-gov-west-1).	29 de janeiro de 2015
Suporte para a região Europa (Frankfurt)	ElastiCache adiciona suporte para a região Europa (Frankfurt) (eu-central-1).	19 de janeiro de 2015

Alteração	Descrição	Alterado em
AWS CloudTrail 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
Posicionamento flexível de nós para Memcached	ElastiCache adiciona suporte para a criação de nós do Memcached em várias zonas de disponibilidade.	23 de julho 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
Descoberta automática PHP	Suporte adicionado para Descoberta automática do PHP versão 5.5.	13 de maio de 2014
Suporte para a Amazon Virtual Private Cloud (VPC) padrão.	Nesta versão, ElastiCache está totalmente integrado com a Amazon Virtual Private Cloud (VPC). Para novos clientes, os clusters de cache são criados em uma Amazon VPC por padrão. Para ter mais informações, consulte Segurança do Amazon VPCs e do ElastiCache .	8 de janeiro de 2013
Suporte a PHP para Descoberta automática de nós de cache	O release inicial da Descoberta automática de nós de cache fornecia suporte para programas Java. Nesta versão, ElastiCache traz suporte à descoberta automática de nós de cache para PHP.	2 de janeiro de 2013

Alteração	Descrição	Alterado em
Suporte para a Amazon Virtual Private Cloud (VPC).	Nesta versão, os ElastiCache clusters podem ser lançados na Amazon Virtual Private Cloud (VPC). Por padrão, clusters de 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
Descoberta automática de nós de cache e nova versão do mecanismo de cache	<p>ElastiCache fornece descoberta automática de nós de cache — a capacidade dos programas clientes determinarem automaticamente todos os nós de cache em um cluster e iniciarem e manterem conexões com todos esses nós.</p> <p>Esta versão também oferece uma nova versão de mecanismo de cache: Memcached versão 1.4.14. Esse novo mecanismo de cache oferece uma melhor capacidade de rebalanceamento de slabs, melhorias significativas de desempenho e escalabilidade e várias correções de bugs. Existem vários novos parâmetros de cache que podem ser configurados. Para ter mais informações, consulte Configuração de parâmetros do mecanismo usando grupos de parâmetros.</p>	28 de novembro de 2012
Novos tipos de nó de cache	Esse release fornece quatro tipos de nó de cache adicionais.	13 de novembro de 2012
Nós de cache reservados	Esse release adiciona suporte para nós de cache reservados.	5 de abril de 2012
Novo guia	Esta é a primeira versão do Amazon ElastiCache User Guide.	22 de agosto de 2011

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Glossário da AWS Referência.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.