Distributed Cache Service

Guia de usuário

Edição 01

Data 2025-01-23

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Marcas registadas e permissões

e outras marcas registadas da Huawei são marcas registadas da Huawei Technologies Co., Ltd. Todos as outras marcas registadas e os nomes registados mencionados neste documento são propriedade dos seus respectivos detentores.

Aviso

Os produtos, os serviços e as funcionalidades adquiridos são estipulados pelo contrato estabelecido entre a Huawei Cloud e o cliente. Os produtos, os serviços e as funcionalidades descritos neste documento, no todo ou em parte, podem não estar dentro do âmbito de aquisição ou do âmbito de uso. Salvo especificação em contrário no contrato, todas as declarações, informações e recomendações neste documento são fornecidas "TAL COMO ESTÃO" sem garantias ou representações de qualquer tipo, sejam expressas ou implícitas.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. Foram feitos todos os esforços na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações contidas neste documento não constituem uma garantia de qualquer tipo, expressa ou implícita.

Índice

1 Antes de começar

1.1 Acessando e usando o DCS

Acessando o DCS

Você pode acessar o Serviço de Cache Distribuído (DCS) do console de gerenciamento baseado na Web ou usando interfaces de programação de aplicativo (as API) RESTful por meio de solicitações HTTPS.

- Usando o console de gerenciamento

  Faça login no **console de gerenciamento** e escolha **Distributed Cache Service** na lista de serviços.

  Para obter detalhes sobre como usar o console de DCS, consulte os capítulos de **Comprando uma instância de DCS** a **Gestão de senhas**.

  Os dados de monitoramento do DCS são registrados pelo Cloud Eye. Para visualizar as métricas de monitoramento ou configurar regras de alarme, acesse o console do Cloud Eye. Para mais detalhes, consulte **Exibindo Métricas**.

  Se você tiver habilitado o Cloud Trace Service (CTS), as operações de instância do DCS serão registradas pelo CTS. Você pode ver o histórico de operações no console CTS. Para mais detalhes, consulte **Operações registradas pelo CTS**.

- Usando as API

  O DCS fornece as API RESTful para você integrar o DCS em seu próprio sistema de aplicativos. Para obter detalhes sobre as API de DCS e chamadas de API, consulte a ***Referência da API do Distributed Cache Service***.

AVISO

1. Todas as funções disponíveis podem ser usadas no console. Algumas funções também podem ser usadas através das API. Para obter mais informações sobre como usar funções por meio das API, consulte ***Referência da API do Distributed Cache Service***.
2. Para obter detalhes sobre as API para monitoramento e auditoria, consulte a documentação do **Cloud Eye** e **Cloud Trace Service**.

Usando o DCS

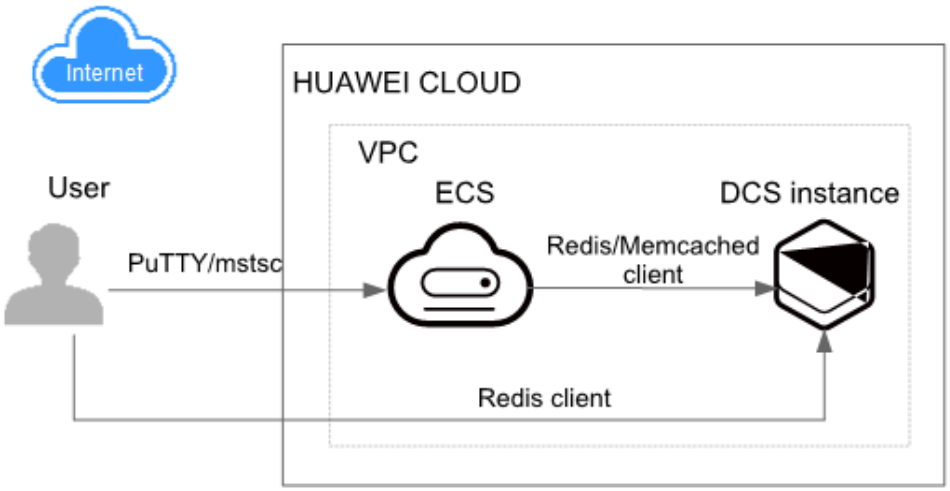
Depois de comprar uma instância DCS, acesse-a referindo-se a **Acessando uma instância do DCS Redis**. Qualquer cliente compatível com o protocolo Redis ou Memcached de código aberto pode acessar uma instância do DCS Redis ou do Memcached. Depois de acessar uma instância do DCS, você pode desfrutar das operações rápidas de leitura/gravação habilitadas pelo DCS.

AVISO

O DCS não envolve informações confidenciais do usuário. Qual, por que, quando e como os dados são processados com a DCS devem estar em conformidade com as leis e regulamentos locais. Se dados confidenciais precisarem ser transmitidos ou armazenados, criptografe os dados antes da transmissão ou armazenamento.

Para obter detalhes sobre como acessar uma instância de DCS, consulte **Figura 1-1**.

Figura 1-1 Acessando uma instância do DCS



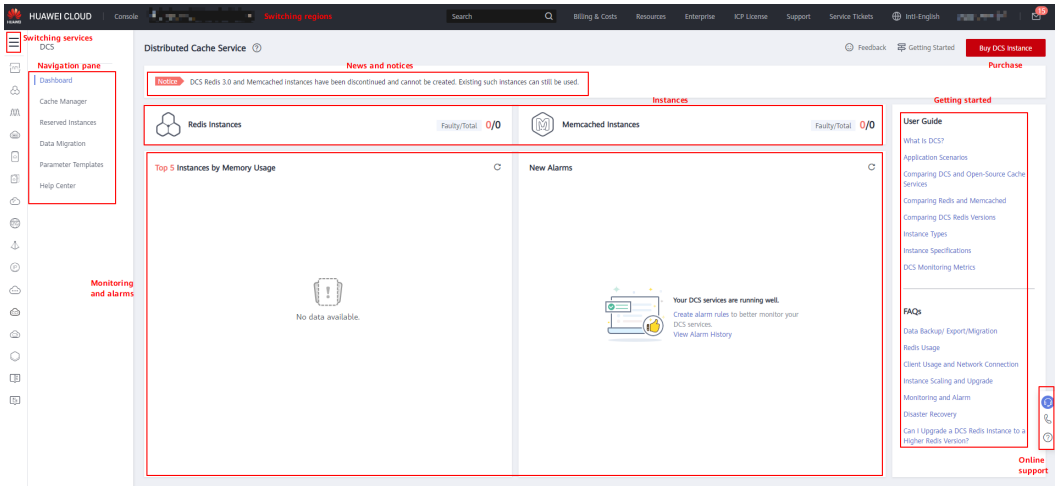
NOTA

- Atualmente, uma instância do DCS pode ser acessada por uma rede interna por meio de um Elastic Cloud Server (ECS) que está na mesma Virtual Private Cloud (VPC) que a instância do DCS.
- Se o acesso público estiver habilitado, uma instância do DCS Redis poderá ser acessada por meio de um endereço IP elástico (EIP) em uma rede pública.

1.2 Usando o Console do DCS

No **Console do DCS**, você pode comprar, usar e manter instâncias do DCS, exibir o status da instância e o uso da memória e buscar suporte online.

Figura 1-2 Console de DCS



- Mudar de regiões

  Você pode mudar para uma região mais próxima da sua aplicação.

- Troca de serviços

  Você pode alternar para consoles de outros serviços, como os consoles VPC e Cloud Eye.

- Criando uma instância

  Clique para comprar instâncias do DCS Redis ou Memcached.

- Painel de Navegação

  Esta área fornece acesso a instâncias de DCS operacionais e migração de dados.

- Notícias e avisos

  Esta área informa-o sobre as últimas funcionalidades disponíveis e ofertas especiais.

- Instâncias

  Esta área exibe o número total de instâncias e o número de instâncias defeituosas do usuário atual.

- Monitoramento e alarmes

  Essa área exibe as instâncias com o maior uso de memória. Para obter detalhes sobre como exibir informações sobre uma instância específica, consulte **Exibindo Detalhes da Instância**.

  Você pode criar regras de alarme para sua instância. Quando um alarme é gerado, você pode lidar com isso imediatamente. Para mais detalhes, consulte **Configurando Regras de Alarme para Métricas Críticas**.

- Introdução

  Ao clicar nessas ligações, você será direcionado para a documentação para saber mais sobre como usar o DCS.

- Suporte online

  Se você tiver alguma dúvida ao usar o DCS, entre em contato com o suporte online.

2 Gerenciamento de permissões

2.1 Criando um Usuário e Concedendo Permissões de DCS

Este capítulo descreve como usar o **IAM** para controle de permissões refinado para seus recursos de DCS. Com o IAM, você pode:

- Crie usuários do IAM para funcionários com base na estrutura organizacional da sua empresa. Cada usuário do IAM terá suas próprias credenciais de segurança para acessar os recursos do DCS.
- Gerencie permissões com base no princípio de menos permissões (PoLP).
- Confie uma conta da HUAWEI CLOUD ou serviço de nuvem para executar O&M eficiente em seus recursos de DCS.

Se sua conta da HUAWEI CLOUD não exigir usuários individuais do IAM, pule este capítulo.

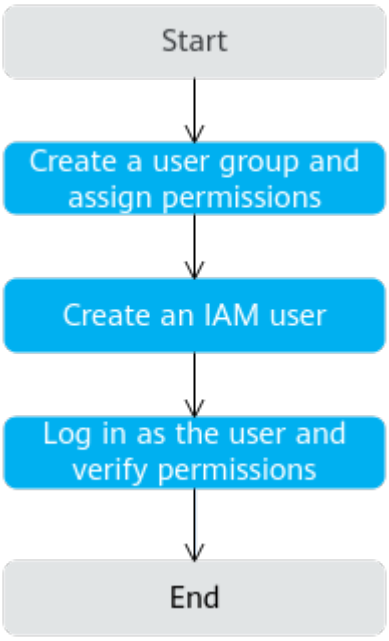
Esta seção descreve o procedimento para conceder a permissão **DCS ReadOnlyAccess** (consulte a **Figura 2-1**) como um exemplo.

Pré-requisitos

Saiba mais sobre as permissões (consulte **Funções e políticas definidas pelo sistema suportadas pelo DCS**) suportados pelo DCS e escolha políticas ou funções de acordo com suas necessidades. Para obter as permissões de outros serviços, consulte **Políticas de permissões**.

Fluxo do Processo

Figura 2-1 Processo de concessão de permissões de DCS



1. **Criar um grupo de usuários e atribuir permissões**.

   Crie um grupo de usuários no console do IAM, e atribua a política de **DCS ReadOnlyAccess** ao grupo.
2. **Criar um usuário do IAM**.

   Crie um usuário no console do IAM e adicione o usuário ao grupo criado em **1**.
3. **Efetue login em** e verifique as permissões.

   Faça login no console do DCS usando o usuário recém-criado e verifique se o usuário só tem permissões de leitura para o DCS.

2.2 Políticas personalizadas do DCS

Políticas personalizadas podem ser criadas para complementar as políticas definidas pelo sistema do DCS. Para as ações que podem ser adicionadas às políticas personalizadas, consulte **Políticas de permissões e ações suportadas**.

Você pode criar políticas personalizadas de uma das seguintes maneiras:

- Editor visual: Selecione serviços de nuvem, ações, recursos e condições de solicitação. Isso não requer conhecimento de sintaxe política.
- JSON: Edite políticas JSON do zero ou com base em uma política existente.

Para obter detalhes, consulte **Criando uma Política Personalizada**. A seção a seguir contém exemplos de políticas personalizadas DCS comuns.

NOTA

Devido ao armazenamento em cache de dados, uma política envolvendo ações do OBS entrará em vigor cinco minutos depois de ser anexada a um usuário, grupo de usuários ou projeto.

Exemplo de Políticas Personalizadas

- Exemplo 1: Permitir que os usuários excluam e reiniciem instâncias de DCS e limpe dados de uma instância

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "
                    dcs:instance:delete
                    dcs:instance:modifyStatus
                 "
            ]
        }
    ]
}
```

- Exemplo 2: Negar a exclusão da instância de DCS

  Uma política com apenas permissões "Negar" deve ser usada em conjunto com outras políticas para entrar em vigor. Se as permissões atribuídas a um usuário contiverem "Permitir" e "Negar", as permissões "Negar" terão precedência sobre as permissões "Permitir".

  Por exemplo, se quiser atribuir todas as permissões da política **DCS FullAccess** a um usuário, exceto excluir instâncias DCS, crie uma política personalizada para negar somente a exclusão de instâncias DCS. Quando você aplica a política de **DCS FullAccess** e a política personalizada negando exclusão de instância DCS, como "Negar" sempre tem precedência sobre "Permitir", o "Negar" será aplicado para essa permissão conflitante. O usuário poderá então executar todas as operações em instâncias de DCS, exceto excluir instâncias de DCS. O seguinte é um exemplo de uma política de negação:

```
{
        "Version": "1.1",
        "Statement": [
                {
                        "Effect": "Deny",
                        "Action": [
                                "dcs:instance:delete"
                        ]
                }
        ]
}
```


3 Comprando uma instância de DCS

3.1 Identificando os requisitos

Antes de comprar uma instância de DCS, identifique suas necessidades:

1. Decida sobre o mecanismo de cache necessário.

   Escolha um mecanismo de cache com base nos requisitos de serviço. O mecanismo de cache não pode ser alterado depois que a instância é criada.

   – Para obter mais informações sobre os mecanismos de cache do Redis e do Memcached, consulte **Visão geral do DCS**.

   – Para obter mais informações sobre as diferenças entre Redis e Memcached, consulte **Comparando Redis e Memcached**.

2. Decida sobre a versão de mecanismo de cache necessária.

   Execute esta etapa se escolher o Redis como mecanismo de cache.

   NOTA

   O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.

   Diferentes versões do Redis têm recursos diferentes. Para obter detalhes, consulte **Comparando versões do DCS Redis**.

3. Decida o tipo de instância.

   O DCS fornece os tipos de instâncias de nó único, principal/em espera, cluster de proxy e cluster do Redis. Para obter detalhes sobre as arquiteturas de instância, consulte **Arquitetura de instância do DCS**.

4. Decida sobre a especificação de instância necessária.

   Cada especificação especifica a memória máxima disponível, o número de conexões e a largura de banda. Para obter detalhes, consulte **Especificações de instâncias do DCS**.

5. Decida sobre a região e se a implantação cross-AZ é necessária.

   Escolha a região mais próxima do seu aplicativo para reduzir a latência.

   Uma região consiste em várias zonas de disponibilidade (as AZ) com fontes de alimentação e redes isoladas fisicamente. As instâncias DCS principal/em espera e de cluster podem ser implantadas nas AZ. Os aplicativos também podem ser implantados nas AZ para obter alta disponibilidade (HA) para dados e aplicativos.

NOTA

- Se uma instância de DCS principal/em espera ou de cluster for implantada nas AZ, as falhas em uma AZ não afetarão os nós de cache nas outras AZ. Isso ocorre porque quando o nó principal está com defeito, o nó de cache em espera se tornará automaticamente o nó principal para fornecer serviços. Tal implantação alcança uma melhor recuperação de desastres.
- A implantação de uma instância de DCS nas AZ reduz ligeiramente a eficiência da rede em comparação com a implantação de uma instância em uma AZ. Portanto, se uma instância de DCS for implantada nas AZ, a sincronização entre os nós de cache principal e em espera será um pouco menos eficiente.

6. Decida se as políticas de backup são necessárias.

   Atualmente, as políticas de backup podem ser configuradas apenas para instâncias de DCS de cluster principal/em espera, de proxy e de cluster do Redis. Para obter detalhes sobre backup e restauração, consulte **Visão geral**.

3.2 Preparando os recursos necessários

Visão geral

Antes de criar uma instância de DCS, prepare os recursos necessários, incluindo uma VPC, sub-rede, grupo de segurança e regras de grupo de segurança. Cada instância de DCS é implantada em uma VPC e vinculada a uma sub-rede e a um grupo de segurança específicos, que fornecem um ambiente de rede virtual isolado e políticas de proteção de segurança que podem ser facilmente configuradas e gerenciadas.

Se você já tiver uma VPC, uma sub-rede e um grupo de segurança, poderá usá-los para todas as instâncias de DCS criadas posteriormente.

Recursos Necessários

A tabela a seguir lista os recursos necessários para uma instância de DCS.

Tabela 3-1 Recursos de dependência de uma instância de DCS

Recurso	Requisito	Operações
VPC e sub-rede	Instâncias de DCS diferentes podem usar as mesmas VPC e sub-redes ou diferentes com base nos requisitos do local. Observe o seguinte ao criar uma VPC e uma sub-rede: ● A VPC e a instância do DCS devem estar na mesma região. ● Mantenha as configurações padrão, a menos que especificado de outra forma.	Para obter detalhes sobre como criar uma VPC e uma sub-rede, consulte **Criando uma VPC**. Se você precisar criar e usar uma nova sub-rede em uma VPC existente, consulte **Criando uma sub-rede para a VPC**.

Recurso	Requisito	Operações
Grupo de segurança **NOTA** Os grupos de segurança são exigidos apenas pelas instâncias do DCS Redis 3.0 e do Memcached.	Instâncias de DCS diferentes podem usar o mesmo grupo de segurança ou grupos de segurança diferentes. Observe o seguinte ao criar um grupo de segurança: ● Defina **Template** como **Custom**. ● Depois que um grupo de segurança for criado, mantenha as regras padrão de entrada e saída. ● Para usar o DCS, você deve adicionar as regras de grupo de segurança descritas em **Tabela 3-2**. Você também pode adicionar outras regras com base nos requisitos do site.	Para obter detalhes sobre como criar um grupo de segurança, consulte **Criando um grupo de segurança**. Para obter detalhes sobre como adicionar regras a um grupo de segurança, consulte **Adicionando uma regra de grupo de segurança**.
(Opcional) EIP **NOTA** Os EIP são compatíveis apenas com instâncias do DCS Redis 3.0.	Se quiser acessar o DCS por meio de uma rede pública, atribua um EIP.	Para obter detalhes sobre como atribuir um EIP, consulte **Atribuindo um EIP**.

Tabela 3-2 Regras de grupos de segurança

Direção	Protocolo	Porta	Origem	Descrição
Entrada	TCP	36379	0.0.0.0/0	Acesse uma instância do DCS Redis (com criptografia SSL ativada) por meio de uma rede pública.
Entrada	TCP	6379	0.0.0.0/0	Acesse uma instância do DCS Redis (com criptografia SSL desabilitada) por meio de uma rede pública.
Entrada	TCP	6379	0.0.0.0/0	Acessar uma instância do DCS Redis em uma rede privada. (A criptografia SSL não é suportada.)
Entrada	TCP	11211	0.0.0.0/0	Acesse uma instância do Memcached DCS por meio de uma rede pública. (A criptografia SSL não é suportada.)

3.3 Comprando uma Instância DCS Redis

Você pode comprar uma ou mais instâncias do DCS Redis com os recursos de computação e o espaço de armazenamento necessários com base nos requisitos de serviço.

NOTA

- O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.
- As versões e os tipos de instância do Redis compatíveis variam de acordo com as regiões.

Pré-requisitos

- Para obter um gerenciamento refinado de seus recursos da HUAWEI CLOUD, crie grupos de usuários e usuários do IAM e conceda permissões especificadas aos usuários. Para mais detalhes, consulte Gerenciamento de permissões.
- Você preparou os recursos necessários.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 Clique em **Buy DCS Instance** no canto superior direito.

Passo 4 Especificar **Billing Mode**.

Passo 5 Selecione a região mais próxima do seu aplicativo para reduzir a latência e acelerar o acesso.

Passo 6 Especifique os seguintes parâmetros de instância com base nas informações coletadas em **Identificando os requisitos**.

1. **Cache Engine**:

   Selecione **Redis**.

2. **Version**:

   Versões do Redis atualmente suportadas: 5.0, 4.0 e 3.0

3. Defina **Instance Type** como **Single-node**, **Master/Standby**, **Proxy Cluster**, **Redis Cluster**, ou **Read/Write splitting**.
4. Defina **CPU Architecture** como **x86** ou **Arm**.
5. Definir **Replicas**. O valor padrão é **2**.

   Esse parâmetro é exibido somente quando você seleciona Redis 4.0 ou Redis 5.0 e o tipo de instância é principal/em espera ou Cluster do Redis.

6. Se **Proxy Cluster** ou **Redis Cluster** estiver selecionado, o parâmetro **Sharding** será exibido.

   Use default: Use as especificações de sharding padrão.

   Customize: Personalize o tamanho de cada estilhaço e selecione as especificações de instância correspondentes.

7. Selecione uma AZ.

   NOTA

   Para acelerar o acesso, implante sua instância e seu aplicativo na mesma AZ.

   Se o tipo de instância for mestre/em espera, Cluster de proxy ou Cluster do Redis, **AZ** torna-se **Primary AZ**, e **Standby AZ** é exibida. Selecione uma AZ para os nós mestre e stand-by da instância.

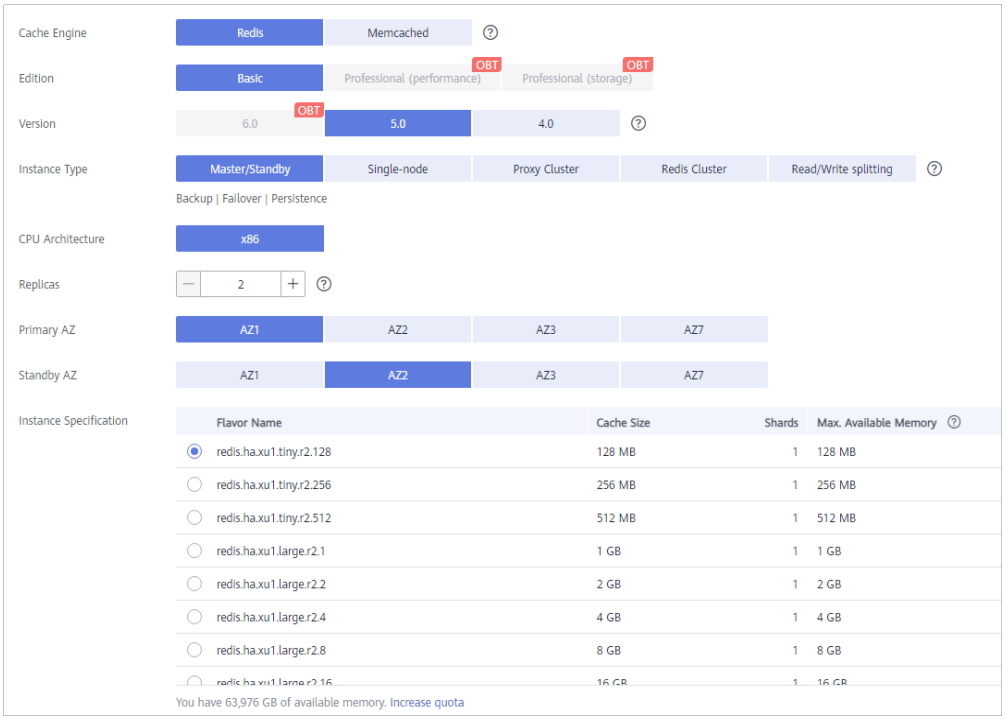
8. **Instance Specification**:

   A cota padrão é exibida no console.

   Para aumentar a quota, clique em **Increase quota** abaixo das especificações. Na página exibida, preencha um formulário de solicitação de cota e clique em **Submit**.

   Figura 3-1 mostra as configurações do parâmetro da instância.

Figura 3-1 Comprando uma instância do DCS Redis



Passo 7 Configurar os parâmetros de rede da instância.

1. Selecione uma VPC e uma sub-rede.
2. Configure o endereço IP.

   As instâncias do Redis Cluster suportam apenas endereços IP atribuídos automaticamente. Os outros tipos de instância suportam endereços IP atribuídos automaticamente e endereços IP especificados manualmente. Você pode especificar manualmente um endereço IP privado para sua instância, conforme necessário.

   Para o Redis 4.0/5.0, você pode especificar uma numeração de porta no intervalo de 1 a 65535. Se nenhuma porta for especificada, a porta padrão 6379 será usada.

   Para o Redis 3.0, não é possível personalizar uma porta. A porta padrão 6379 será usada.

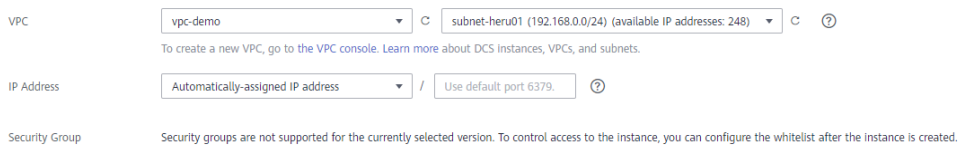
3. Selecione um grupo de segurança.

Um grupo de segurança é um conjunto de regras que controlam o acesso aos ECS. Ele fornece políticas de acesso para os ECS mutuamente confiáveis com os mesmos requisitos de proteção de segurança na mesma VPC.

O DCS for Redis 4.0/5.0 é baseado no VPC Endpoint e não requer grupos de segurança.

Se a porta 6379 não estiver ativada para o grupo de segurança selecionado, a caixa de seleção **Enable port 6379** será exibida e selecionada por padrão, indicando que, após a criação da instância, a porta 6379 será habilitada para o grupo de segurança selecionado. Se a porta 6379 não estiver ativada para o grupo de segurança selecionado, as conexões com a instância podem falhar.

Figura 3-2 Configurando parâmetros de rede da instância



Passo 8 Defina a senha da instância.

- Selecione **Yes** ou **No** para **Password Protected**.

  NOTA

  – O acesso sem senha acarreta riscos de segurança. Tenha cuidado ao selecionar este modo.
  – Para habilitar o acesso público para uma instância do DCS Redis 3.0, selecione o modo protegido por senha e defina uma senha.
  – Depois de criar uma instância do DCS Redis para ser acessada no modo sem senha, você pode definir uma senha para ela usando a função de redefinição de senha. Para mais detalhes, consulte **Alteração das configurações de senha para instâncias do DCS Redis**.

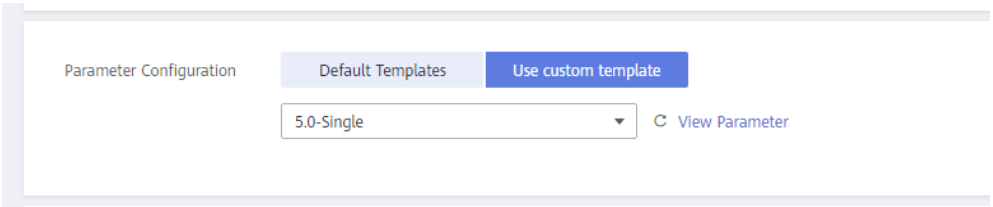
- **Password** e **Confirm Password**: Esses parâmetros indicam a senha de acesso à instância do DCS Redis e são exibidos somente quando **Password Protected** (Protegido por senha) estiver definido como **Yes**.

  NOTA

  Por motivos de segurança, se o acesso sem senha estiver desativado, o sistema solicitará que você insira uma senha específica da instância quando estiver acessando a instância do DCS Redis. Mantenha a senha da instância segura e altere-a periodicamente.

Passo 9 Configurar **Parameter Configuration**.

Você pode selecionar **Default Templates** ou **Use custom template**.



NOTA

- Na página de criação de instância, os modelos de parâmetro padrão são usados por padrão.
- Você pode selecionar um modelo personalizado somente para instâncias do Redis. A versão do mecanismo de cache selecionado e o tipo de instância devem corresponder aos do modelo.

Passo 10 Especifique a duração e a quantidade de instâncias necessárias para o faturamento anual/mensal.

Passo 11 Informe um nome de instância e selecione um projeto da empresa.

Quando você cria apenas uma ocorrência por vez, o valor de **Name** pode conter de 4 a 64 caracteres. Quando você cria mais de uma ocorrência por vez, o valor de **Name** pode conter de 4 a 56 caracteres. Essas instâncias são nomeadas no formato de "*name-n*", no qual *n* começa de 000 e é incrementado de 1. Por exemplo, se você criar duas instâncias e definir **Name** como **dcs_demo**, as duas instâncias serão nomeadas respectivamente como **dcs_demo-000** e **dcs_demo-001**.

Passo 12 Clique em **More Settings** para exibir mais configurações, incluindo política de backup e renomeação de comandos críticos.

1. Insira uma descrição da instância.
2. Especifique a política de backup.

   Esse parâmetro é exibido somente quando o tipo de instância é principal/em espera ou cluster. Para obter mais informações sobre como configurar uma política de backup, consulte **Backup e restauração de instâncias**.
3. Renomear comandos críticos.

   Se Redis 4.0/5.0 estiver selecionado, o parâmetro **Command Renaming** será exibido. Atualmente, você só pode renomear os comandos **COMMAND**, **KEYS**, **FLUSHDB**, **FLUSHALL** e **HGETALL**.
4. Especifique a janela de manutenção.

   Escolha uma janela para que o pessoal de O&M da DCS realize a manutenção em sua instância. Você será contatado antes que qualquer atividade de manutenção seja realizada.
5. Adicione uma tag.

   As tags são usadas para identificar os recursos da nuvem. Quando você tem muitos recursos de nuvem do mesmo tipo, pode usar tags para classificar os recursos de nuvem por dimensão (por exemplo, por uso, proprietário ou ambiente).
   - Se você tiver criado tags predefinidas, selecione um par predefinido de chave e valor de tag. Clique em **View predefined tags**. Na consola do Serviço de Gestão de Etiquetas (TMS), veja as etiquetas predefinidas ou crie novas etiquetas.
   - Você também pode adicionar uma tag inserindo a chave e o valor da tag. Para obter detalhes sobre os requisitos de nomeação de tags, consulte **Gerenciando Tags**.

Passo 13 Clique em **Next**.

A página exibida mostra as informações da instância que você especificou.

Passo 14 Confirme as informações da instância e submeta a solicitação.

Passo 15 Retorne à página **Cache Manager** para exibir e gerenciar suas instâncias de DCS.

----Fim

3.4 Comprando uma instância do DCS Memcached (indisponível em breve)

Você pode comprar uma ou mais instâncias do DCS Memcached com os recursos de computação e o espaço de armazenamento necessários com base nos requisitos de serviço.

NOTA

O DCS para Memcached está prestes a ficar indisponível e não é mais vendido em algumas regiões. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.

Comprando uma instância do Memcached do DCS

Passo 1 Efetue login no **console do DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique em **Buy DCS Instance** no canto superior direito.

Passo 5 Especificar **Billing Mode**.

Passo 6 Selecione a região mais próxima do seu aplicativo para reduzir a latência e acelerar o acesso.

Passo 7 Especifique os seguintes parâmetros de instância com base nas informações coletadas em **Identificando os requisitos**.

1. Defina **Cache Engine** como **Memcached**.
2. Defina **Instance Type** como **Single-node** ou **Master/Standby**.
3. Selecione uma AZ.

   NOTA

   Para acelerar o acesso, implante sua instância e seu aplicativo na mesma AZ. Para garantir a confiabilidade dos dados, implante-os nas diferentes AZ.

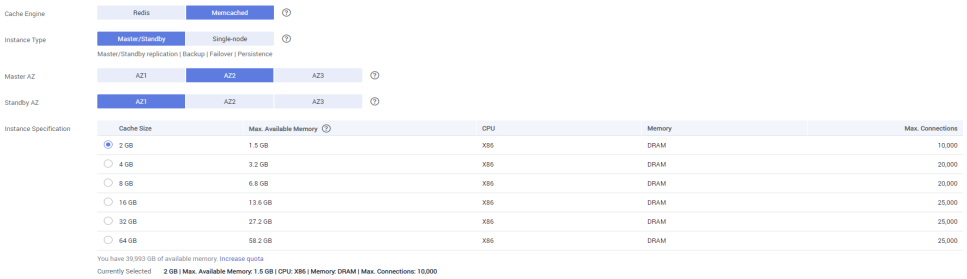
   Se o tipo de instância for mestre/em espera, **Standby AZ** será exibida. Selecione uma AZ stand-by para o nó stand-by da instância.
4. Especificar **Instance Specification**.

   A cota padrão é exibida no console.

   Para aumentar a quota, clique em **Increase quota** abaixo das especificações. Na página exibida, preencha um formulário de solicitação de cota e clique em **Submit**.

   Figura 3-3 mostra as configurações do parâmetro da instância.

Figura 3-3 Comprando uma instância do Memcached do DCS



Passo 8 Configurar os parâmetros de rede da instância.

1. Para **VPC**, selecione uma VPC, sub-rede e especifique o endereço IP.

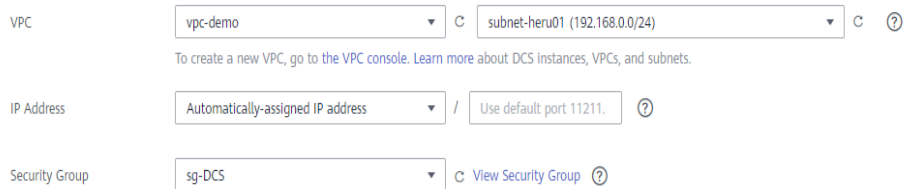
   Você pode optar por obter um endereço IP atribuído automaticamente ou especificar manualmente um endereço IP disponível na sub-rede selecionada.

2. Selecione um grupo de segurança.

   Um grupo de segurança é um conjunto de regras que controlam o acesso aos ECS. Ele fornece políticas de acesso para os ECS mutuamente confiáveis com os mesmos requisitos de proteção de segurança na mesma VPC.

   Se a porta 11211 não estiver habilitada para o grupo de segurança selecionado, a caixa de seleção **Enable port 11211** será exibida e selecionada por padrão, indicando que, após a criação da instância, a porta 11211 será ativada para o grupo de segurança selecionado. Se a porta 11211 não estiver habilitada para o grupo de segurança selecionado, as conexões com a instância podem falhar.

Figura 3-4 Configurando parâmetros de rede da instância



Passo 9 Defina a senha da instância.

- Selecione **Yes** ou **No** para **Password Protected**.

  NOTA

  - O acesso sem senha acarreta riscos de segurança. Tenha cuidado ao selecionar este modo.
  - Depois de criar uma instância do Memcached DCS no modo protegido por senha, você pode redefinir a senha ou alterá-la para o modo sem senha. Para mais detalhes, consulte **Alterando as Configurações de Senha para Instâncias de Memcached do DCS**.
  - Se o acesso sem senha estiver desativado, as instâncias do Memcached do DCS devem ser acessadas usando o protocolo binário do Memcached e por meio da autenticação SASL.

- Nome de usuário necessário para acessar a nova instância do DCS.

  NOTA

  Esse parâmetro é exibido somente quando **Password Protected** é definido como **Yes**.

- **Password** e **Confirm Password**: Esses parâmetros indicam a senha de acesso à instância do Memcached DCS e são exibidos somente quando **Password Protected** é definido como **Yes**.

> **NOTA**
>
> Por motivos de segurança, se o acesso sem senha estiver desabilitado, o sistema solicitará que você insira uma senha específica da instância quando estiver acessando a instância do Memcached DCS. Mantenha a senha da instância segura e altere-a periodicamente.

Passo 10 Especifique a duração e a quantidade necessárias.

Passo 11 Informe um nome de instância e selecione um projeto corporativo.

Quando você cria apenas uma instância por vez, o valor de **Name** pode conter de 4 a 64 caracteres. Quando você cria mais de uma ocorrência por vez, o valor de **Name** pode conter de 4 a 56 caracteres. Essas instâncias são nomeadas no formato de "*name-n*", no qual *n* começa de 000 e é incrementado de 1. Por exemplo, se você criar duas instâncias e definir **Name** como **dcs_demo**, as duas instâncias serão nomeadas respectivamente como **dcs_demo-000** e **dcs_demo-001**.

Passo 12 Clique em **More Settings** para exibir mais configurações, incluindo tags de instância e política de backup.

1. Insira uma descrição da instância.
2. Especifique a política de backup.

   Esse parâmetro é exibido somente quando o tipo de instância é mestre/em espera. Para obter mais informações sobre como configurar uma política de backup, consulte **Backup e restauração de instâncias**.
3. Especifique a janela de manutenção.

   Especifique um período para o pessoal do DCS O&M manter sua instância. Por exemplo, se você escolher 02:00-03:00, os nós da instância serão mantidos durante esse período.
4. Adicione uma tag.

   As tags são usadas para identificar os recursos da nuvem. Quando você tem muitos recursos de nuvem do mesmo tipo, pode usar tags para classificar os recursos de nuvem por dimensão (por exemplo, por uso, proprietário ou ambiente).

   – Se você tiver criado tags predefinidas, selecione um par predefinido de chave e valor de tag. Clique em **View predefined tags**. Na consola do Serviço de Gestão de Etiquetas (TMS), veja as etiquetas predefinidas ou crie novas etiquetas.

   – Você também pode criar novas tags especificando **Tag key** e **Tag value**.

   Até 10 tags podem ser adicionadas a cada instância de DCS. Para obter detalhes sobre os requisitos de tags, consulte **Gerenciando Tags**.

Passo 13 Clique em **Next**.

A página exibida mostra as informações da instância que você especificou.

Passo 14 Confirme as informações da instância e clique em **Submit**.

Passo 15 Retorne à página **Cache Manager** para exibir e gerenciar suas instâncias de DCS.

1. Leva de 5 a 15 minutos para criar uma instância de DCS.
2. Depois que uma instância de DCS é criada com êxito, ela entra no estado **Running** por padrão.

----Fim

4 Acessando uma instância do DCS Redis

4.1 Restrições

Você pode acessar uma instância do DCS por meio de qualquer cliente do Redis. Para obter detalhes sobre os clientes da Redis, consulte **o site oficial da Redis**.

NOTA

O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.

- Para acessar uma instância do DCS Redis por meio de um cliente em um ECS na mesma VPC da instância, observe que:
  - Um ECS e uma instância de DCS só podem se comunicar quando pertencerem à mesma VPC.
    - Redis 3.0 A instância e o ECS devem ser configurados com o mesmo grupo de segurança ou usar grupos de segurança diferentes, mas podem se comunicar entre si conforme configurado pelas regras do grupo de segurança.
    - Redis 4.0/5.0: O endereço IP do ECS deve estar na lista de permissões da instância do DCS.
  - Se a instância do ECS e do DCS Redis estiverem nas VPC diferentes, estabeleça uma conexão de peering de VPC para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **O DCS oferece suporte ao acesso entre VPC?**
- Antes de acessar uma instância do DCS Redis 3.0 em redes públicas, certifique-se de que:

  As regras do grupo de segurança foram configuradas corretamente para a instância. Se a encriptação SSL estiver desactivada, permita o acesso público através da porta 6379. Se a encriptação SSL estiver activada, permita o acesso público através da porta 36379. Para obter detalhes, consulte **Como configurar um grupo de segurança?**
- Se o cliente e a instância do DCS Redis não estiverem na mesma região, o nome de domínio da instância não poderá ser resolvido entre regiões e a instância não poderá ser acessada em seu endereço de nome de domínio. Você pode mapear manualmente o nome de domínio para o endereço IP no arquivo **hosts** ou acessar a instância em seu endereço IP.

4.2 Acesso público a uma instância do DCS Redis 3.0 (Indisponível)

4.2.1 Passo 1: Verifique se o acesso público é suportado

Você pode acessar uma instância do DCS Redis 3.0 em redes públicas. Em comparação com o acesso intra-VPC, o acesso público pode trazer perda de pacotes, jitter e maior latência. Portanto, é aconselhável habilitar o acesso público somente durante as fases de desenvolvimento e teste do serviço.

Antes de se conectar a uma instância do DCS por redes públicas, verifique se a instância oferece suporte ao acesso público.

- Redis 3.0

  Atualmente, **apenas as instâncias do DCS Redis 3.0 oferecem suporte ao acesso público**. Você pode habilitar ou desabilitar o acesso público.

- Redis 4.0 e Redis 5.0

  O acesso público não é suportado pelas instâncias do DCS Redis 4.0 e 5.0 Se o acesso público for necessário para uma instância de cluster de nó único, principal/em espera ou proxy, use o Nginx para redirecionar conexões por meio de um ECS configurado com a mesma VPC e grupo de segurança da instância DCS. Para obter detalhes, consulte **Usando o Nginx para acessar instâncias do DCS Redis 4.0 ou 5.0 em redes públicas**.

  As instâncias do Redis Cluster não podem ser acessadas em redes públicas.

- Memcached

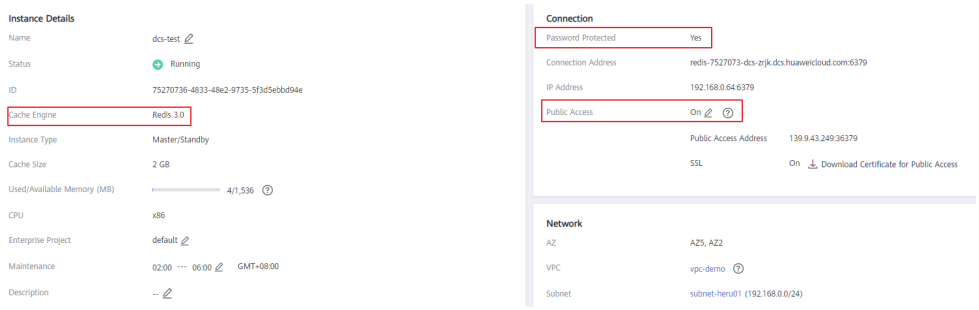
  O acesso público não é suportado pelas instâncias do Memcached do DCS. O ECS que serve como cliente e a instância do DCS que o cliente acessará devem pertencer à mesma VPC. Na fase de desenvolvimento e depuração de aplicativos, você também pode usar um agente SSH para acessar instâncias do DCS Memcached no ambiente local.

Procedimento

Na página **Basic Information** da instância, verifique as seguintes definições de parâmetros:

- **Cache Engine**: Deve ser **Redis 3.0**. Caso contrário, o acesso à rede pública não é suportado.
- **Password Protected**: Deve ser **Yes**. Caso contrário, ative a proteção por senha para a instância, referindo-se a **Perguntas Frequentes**.
- **Public Access**: Deve ser **On**. Se não, habilite o acesso público fazendo referência a **Passo 2: Habilitar acesso público para uma instância do DCS Redis**.

Figura 4-1 Verificando a versão do mecanismo de cache, a proteção por senha e o acesso público



Perguntas Frequentes

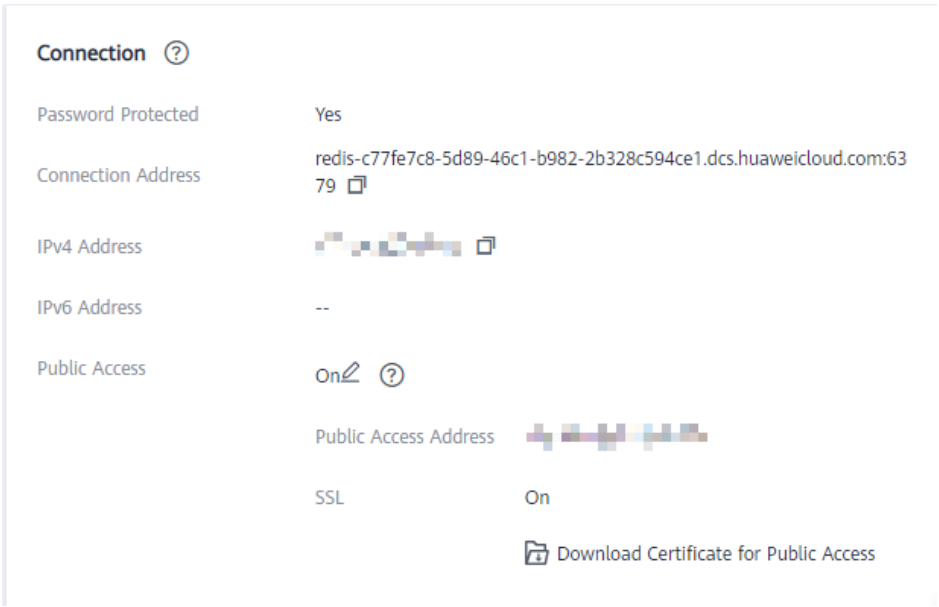
- O que posso fazer se o botão de acesso público ficar esmaecido quando a instância não estiver protegida por senha?

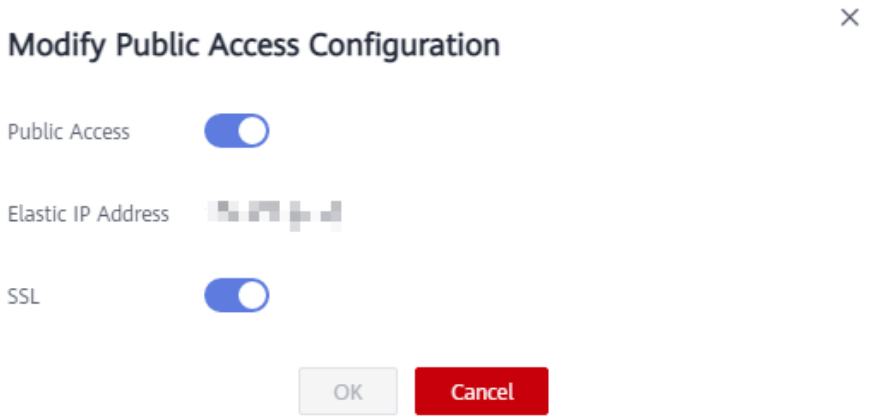
  No canto superior direito da página **Basic Information**, escolha **More** > **Reset Password**. Depois que a senha é redefinida, o parâmetro **Password Protected** muda para **Yes**. O botão de acesso público pode ser clicado agora.

- Como faço para desabilitar a criptografia SSL quando o acesso público foi habilitado?

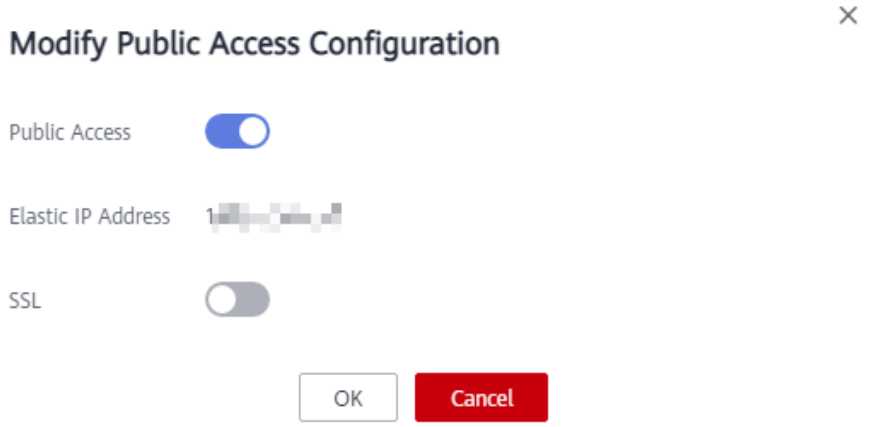
  A encriptação SSL é activada por predefinição quando activa o acesso público. Para desativar a criptografia SSL, execute as seguintes etapas:

  a. Abra a página para configurar o acesso público.





b. Desabilite a criptografia SSL e clique em **OK**.



c. Na área **Connection** na página de detalhes da instância, o **SSL** está desabilitado.

4.2.2 Passo 2: Habilitar acesso público para uma instância do DCS Redis

Se o acesso público tiver sido ativado para a instância, ignore esta seção.

Se o acesso público não estiver ativado, siga as instruções nesta seção. Você pode ativar ou desativar a criptografia SSL ao habilitar o acesso público.

NOTA

- Antes de acessar uma instância DCS por meio de uma rede pública (com criptografia SSL), baixe um certificado de CA para verificar o certificado da instância para fins de segurança.
- Ao acessar uma instância DCS por meio de uma rede pública (sem criptografia SSL), acesse o EIP e a porta 6379 da instância. Você não precisa baixar certificados ou instalar o Stunnel no seu cliente.
- Recomendamos que você habilite o SSL para criptografar os dados transmitidos entre o cliente do Redis e a instância do DCS para evitar vazamento de dados.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome da instância do DCS Redis que deseja configurar. Uma página com informações básicas sobre a instância do DCS é exibida.

Passo 5 Clique em no lado direito de **Public Access**.

Passo 6 Clique em para ativar o acesso público.

Passo 7 Selecione um EIP na lista suspensa **Elastic IP Address**.

Se nenhum EIP estiver disponível, clique em **View Elastic IP** para criar um EIP no console da rede. Depois que um EIP for criado, clique no botão Atualizar à direita de **Elastic IP Address** para selecionar o novo EIP.

Passo 8 (Opcional) Habilite ou desabilite o SSL conforme necessário.

Recomendamos que você habilite o SSL para criptografar os dados transmitidos entre o cliente do Redis e a instância do DCS para evitar vazamento de dados.

Passo 9 Clique em **OK**.

Demora de 1 a 2 minutos para permitir o acesso público.

Você será automaticamente redirecionado para a página **Background Tasks**, onde o progresso da tarefa atual é exibido. Se o status da tarefa for **Succeeded**, o acesso público será ativado com êxito.

----Fim

4.2.3 Passo 3: Acessar uma instância do DCS Redis no Windows

Esta seção descreve como acessar uma instância do DCS Redis 3.0 em uma rede pública usando o redis-cli no Windows.

O acesso público ajuda o pessoal de P&D a estabelecer um ambiente local para desenvolvimento ou teste, melhorando a eficiência do desenvolvimento. No entanto, no ambiente de produção (ambiente oficial), acesse uma instância do DCS Redis por meio de uma VPC para garantir um acesso eficiente.

Pré-requisitos

Antes de usar o redis-cli para acessar uma instância do DCS Redis em uma rede pública, verifique se:

- A versão da instância é o Redis 3.0 e o acesso público foi habilitado.

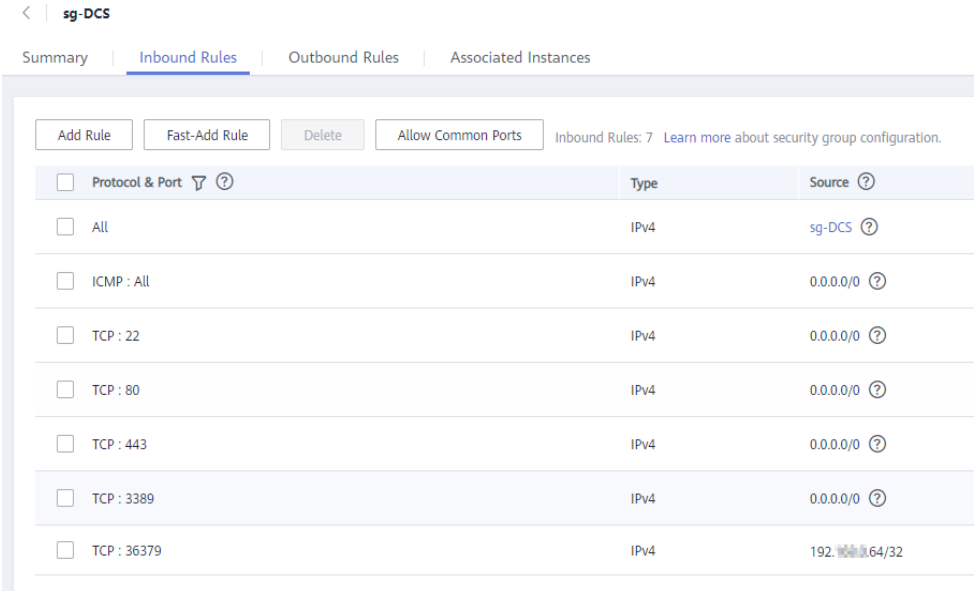
- Se os certificados forem necessários para acessar a instância do DCS, faça download do certificado na página de detalhes da instância do DCS. Para mais detalhes, consulte **Exibindo Detalhes da Instância**.

Conectando-se ao Redis com criptografia SSL

Passo 1 Assegure-se de que a regra do grupo de segurança permita o acesso público através da porta 36379.

Quando a encriptação SSL estiver activada, permita o acesso público através da porta 36379 e instale o cliente Stunnel.

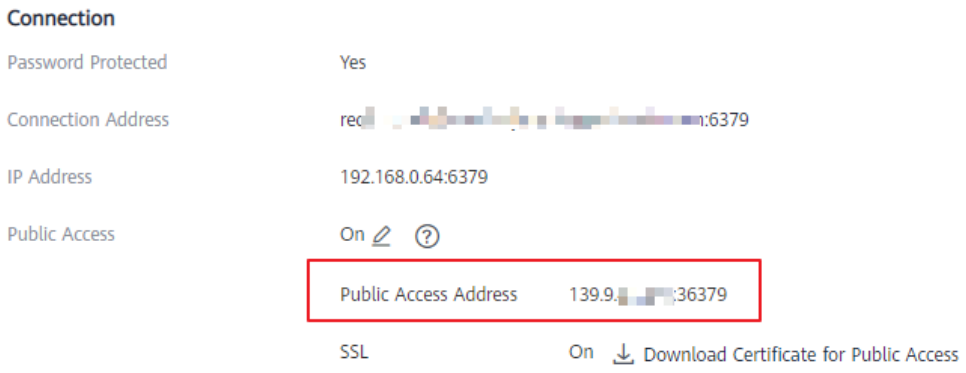
Figura 4-2 Regra de grupo de segurança (porta 36379)



Passo 2 Obtenha o endereço de acesso público e os certificados da instância na página **Basic Information** da instância.

- O endereço de acesso público é exibido na seção **Connection**.
- Os certificados podem ser baixados clicando em **Download Certificate for Public Access** na seção **Connection**. Após a descompactação, você obterá **dcs-ca.cer** (o certificado de chave pública em formato binário) e **dcs-ca-bundle.pem** (o arquivo de certificado em formato de texto).

Figura 4-3 Visualizando o endereço de acesso público (SSL habilitado; porta 36379)



Passo 3 Baixe o pacote de instalação mais recente do Windows Stunnel (por exemplo, **stunnel-5.44-win32-installer.exe**) do **https://www.stunnel.org/downloads.html** para o dispositivo Windows local.

Passo 4 Execute o programa de instalação do Stunnel e instale o cliente do Stunnel.

Passo 5 Configure o cliente Stunnel: Clique em com o botão direito do mouse na barra de tarefas e escolha **Edit Configuration**. Adicione a seguinte configuração e, em seguida, salve e saia.

```
[redis-client]
client = yes
CAfile = D:\tmp\dcs\dcs-ca.cer
accept = 8000
connect = {public access address}
```

Na configuração:

- **client**: indica Stunnel. O valor fixo é **yes**.
- **CAfile**: especifica um certificado CA, que é opcional. Se um certificado CA for necessário, baixe e descompacte o certificado **dcs-ca.cer** conforme instruído em **Passo 2**. Se não for necessário, exclua este parâmetro.
- **accept**: especifica o número da porta de escuta definida pelo usuário do Stunnel. Especifique esse parâmetro ao acessar uma instância de DCS usando um cliente Redis.
- **connect**: especifica o endereço de serviço e o número da porta do Stunnel. Defina esse parâmetro como o endereço de acesso público da instância obtido em **Passo 2**.

Quando a encriptação SSL está activada, a configuração é semelhante à seguinte:

```
[redis-client]
client = yes
CAfile = D:\tmp\dcs\dcs-ca.cer
accept = 8000
connect = 49.**.**.211:36379
```

Passo 6 Clique em com o botão direito do mouse na barra de tarefas e escolha **Reload Configuration**.

Passo 7 Abra a ferramenta CLI **cmd.exe** e execute o seguinte comando para verificar se 127.0.0.1:8000 está sendo escutado:

netstat -an |find "8000"

Suponha que a porta **8000** esteja configurada como a porta de escuta no cliente.

Se **127.0.0.1:8000** for exibido no resultado retornado e seu status for **LISTENING**, o cliente Stunnel está sendo executado corretamente. Quando o cliente do Redis se conecta ao endereço **127.0.0.1:8000**, o Stunnel encaminhará as solicitações para a instância do DCS Redis.

Passo 8 Acesse a instância do DCS Redis.

1. Obter e descompactar o pacote de instalação do cliente Redis.

   O pacote de instalação do cliente Windows Redis pode ser baixado **aqui**

2. Abra a ferramenta CLI **cmd.exe** e execute comandos para ir para o diretório onde o pacote de instalação do cliente Redis descompactado foi salvo.

   Por exemplo, para ir para o diretório **D:\redis-64.3.0.503**, execute os seguintes comandos:

   D:

   cd D:\redis-64.3.0.503

3. Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

   redis-cli -h 127.0.0.1 -p 8000 -a *<password>*

   ⚠ CUIDADO

   No comando anterior:

   – O endereço a seguir **-h** indica o endereço do cliente Stunnel, que é **127.0.0.1**.

   – A porta a seguir **-p** é a porta de escuta do cliente Stunnel, que foi configurada no campo **accept** em **Passo 5**. **8000** é usado um exemplo aqui.

   Não use o endereço de acesso público e a porta exibidos no console para os parâmetros **-h** e **-p**.

   <password> indica a senha usada para fazer logon na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

   Você acessou a instância com êxito se a seguinte saída do comando for exibida:

   ```
   127.0.0.1:8000>
   ```

   Insira **info** e as informações da instância do DCS serão retornadas. Se nenhuma informação for retornada ou a conexão for interrompida, clique com o botão direito do mouse no ícone Stunnel na barra de tarefas e escolha **Show Log Window** no menu de atalho para mostrar logs de Stunnel para análise de causa.

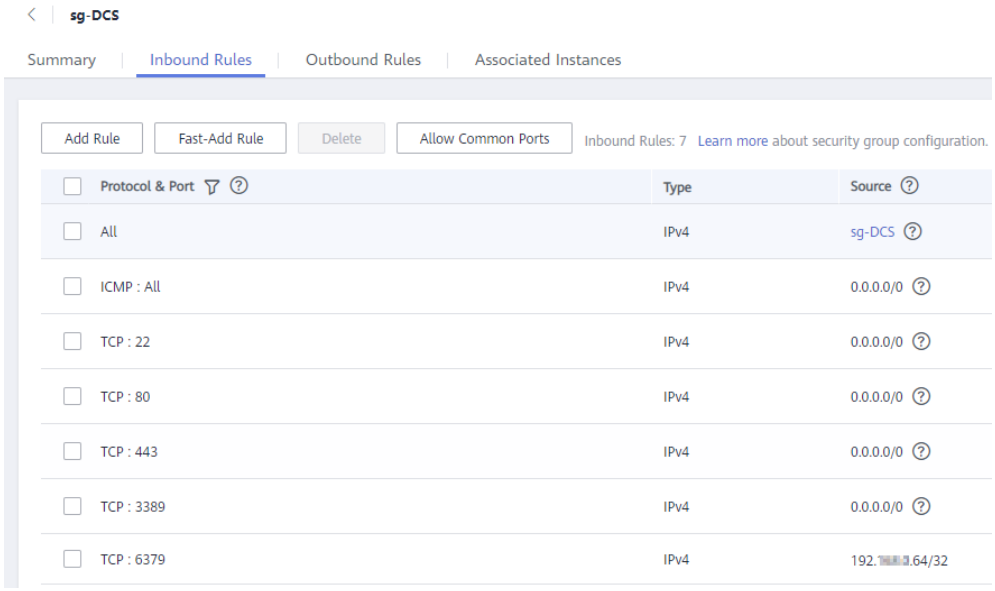
----Fim

Conectando-se ao Redis sem criptografia SSL

Passo 1 Assegure-se de que a regra do grupo de segurança permita o acesso público através da porta 6379.

Quando a criptografia SSL é desabilitada, o endereço de acesso público da instância pode ser acessado somente se o acesso pela porta 6379 for permitido.

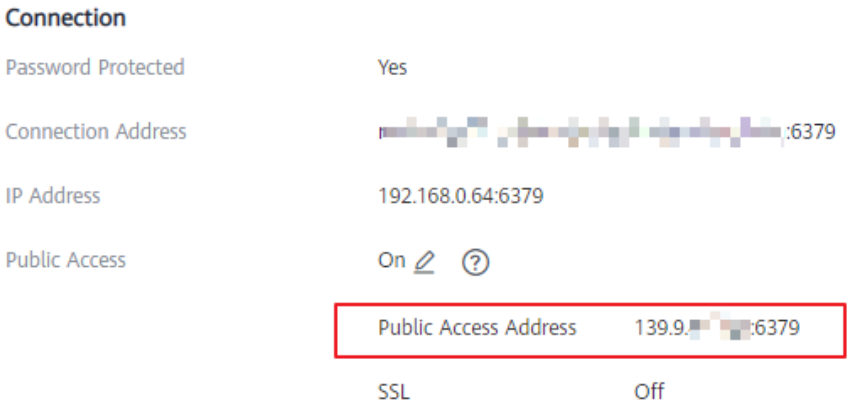
Figura 4-4 Regra de grupo de segurança (porta 6379)



Passo 2 Obtenha o endereço de acesso público da instância.

O endereço de acesso público é exibido na seção **Connection**.

Figura 4-5 Exibindo o endereço de acesso público (SSL desativado; porta 6379)



Passo 3 Obter e descompactar o pacote de instalação do cliente Redis.

O pacote de instalação do cliente Windows Redis pode ser baixado **aqui**

Passo 4 Abra a ferramenta CLI **cmd.exe** e execute comandos para ir para o diretório onde o pacote de instalação do cliente Redis descompactado foi salvo.

Por exemplo, para ir para o diretório **D:\redis-64.3.0.503**, execute os seguintes comandos:

D:

cd D:\redis-64.3.0.503

Passo 5 Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

redis-cli -h *{public network access IP}* **-p 6379 -a** *<password>*

Neste comando, *{public network access IP}* indica o endereço IP da instância do DCS Redis obtida em **Passo 2**. *<password>* indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Você acessou a instância com êxito se a seguinte saída do comando for exibida:

```
139.**.**.175:6379>
```

Insira **info** e as informações da instância do DCS serão retornadas.

----Fim

Solução de problemas

- **Symptom**: "Erro: A redefinição de conexão por peer é exibida ou uma mensagem é exibida indicando que o host remoto fecha forçosamente uma conexão existente.

  Possible cause 1: O grupo de segurança está configurado incorretamente. Você precisa habilitar a porta **36379** ou **6379**.

  Possible cause 2: A criptografia SSL foi habilitada, mas o Stunnel não está configurado durante a conexão. O endereço IP exibido no console foi usado para conexão. Neste caso, siga rigorosamente as instruções fornecidas em **Conectando-se ao Redis com criptografia SSL**.
- Para obter mais informações sobre falhas de conexão do Redis, consulte **Troubleshooting de Falhas de Conexão do Redis** .

4.2.4 Passo 3: Acessar uma instância do DCS Redis no Linux

Esta seção descreve como acessar uma instância do DCS Redis 3.0 em uma rede pública usando o redis-cli no Linux.

O acesso público ajuda o pessoal de P&D a estabelecer um ambiente local para desenvolvimento ou teste, melhorando a eficiência do desenvolvimento. No entanto, no ambiente de produção (ambiente oficial), acesse uma instância do DCS Redis por meio de uma VPC para garantir um acesso eficiente.

Pré-requisitos

Antes de usar o redis-cli para acessar uma instância do DCS Redis em uma rede pública, verifique se:

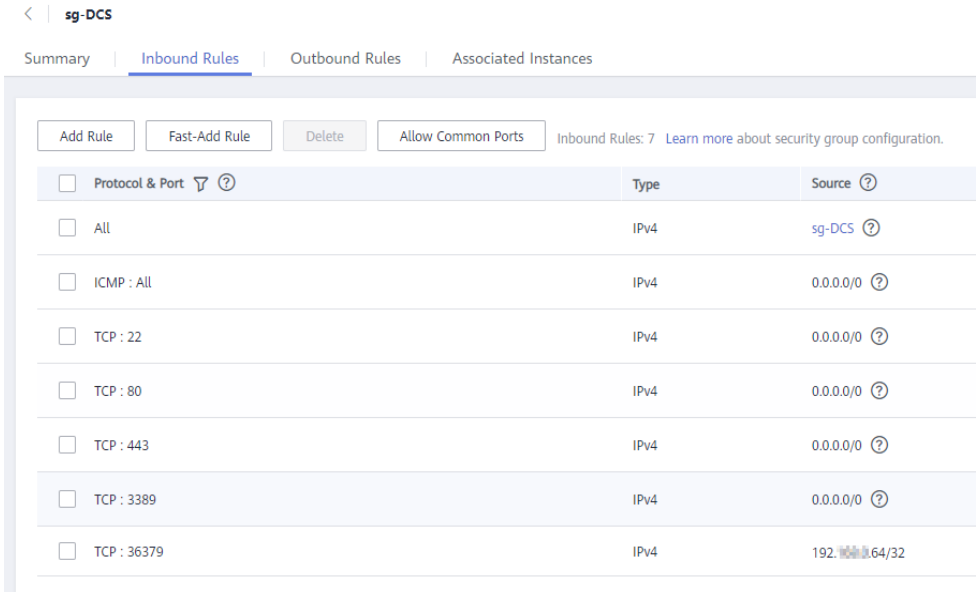
- A versão da instância é o Redis 3.0 e o acesso público foi habilitado.
- Se os certificados forem necessários para acessar a instância do DCS, faça download do certificado na página de detalhes da instância do DCS. Para mais detalhes, consulte **Exibindo Detalhes da Instância**.

Conectando-se ao Redis com criptografia SSL

Passo 1 Assegure-se de que a regra do grupo de segurança permita o acesso público através da porta 36379.

Quando a encriptação SSL estiver activada, permita o acesso público através da porta 36379. Assegure-se de que o cliente do Stunnel esteja instalado.

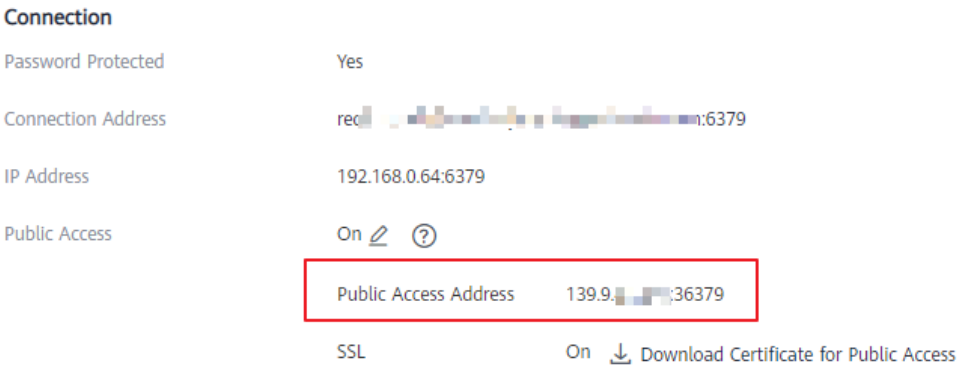
Figura 4-6 Regra de grupo de segurança (porta 36379)



Passo 2 Obtenha o endereço de acesso público e os certificados da instância na página **Basic Information** da instância.

- O endereço de acesso público é exibido na seção **Connection**.
- Os certificados podem ser baixados clicando em **Download Certificate for Public Access** na seção **Connection**. Após a descompactação, você obterá **dcs-ca.cer** (o certificado de chave pública em formato binário) e **dcs-ca-bundle.pem** (o arquivo de certificado em formato de texto).

Figura 4-7 Visualizando o endereço de acesso público (SSL habilitado; porta 36379)



Passo 3 Efetue login no dispositivo Linux local.

Passo 4 Instale o cliente Stunnel.

Utilize um dos seguintes métodos para instalar o Stunnel.

NOTA

Os métodos de instalação **apt** e **yum** são recomendados. Qualquer SO de Linux comum deve suportar pelo menos um desses métodos de instalação.

- Método **apt-get**:

  O **apt-get** é usado para gerenciar pacotes de software DEB e aplicável a sistemas operacionais Debian como o Ubuntu. Execute o seguinte comando para instalar o Stunnel:

  apt install stunnel ou **apt-get install stunnel**

  Se você não conseguir encontrar o Stunnel depois de executar o comando, execute o comando **apt update** para atualizar a configuração e, em seguida, instale o Stunnel novamente.

- Método do **yum**:

  O **yum** é usado para gerenciar pacotes de software RPM e aplicável aos SO como Fedora, CentOS e Red Hat. Execute o seguinte comando para instalar o Stunnel:

  yum install stunnel

Passo 5 Abra o arquivo de configuração Stunnel **stunnel.conf**.

- Se o Stunnel for instalado usando o **apt-get**, o arquivo de configuração é armazenado no diretório **/etc/stunnel/stunnel.conf** por padrão.

  Se esse diretório não existir ou não existir nenhum arquivo de configuração nele, adicione um diretório ou arquivo de configuração.

- Se o Stunnel for instalado usando o **yum**, o arquivo de configuração é armazenado no diretório **/usr/local/stunnel/stunnel.conf** por padrão.

  Se esse diretório não existir ou não existir nenhum arquivo de configuração nele, adicione um diretório ou arquivo de configuração.

  NOTA

  - Se você não tiver certeza de onde armazenar o arquivo de configuração, digite o comando **stunnel** após a instalação para exibir o diretório para armazenar o arquivo de configuração.
  - O arquivo de configuração pode ser armazenado em qualquer diretório. Especifique este arquivo de configuração ao iniciar o Stunnel.

Passo 6 Adicione o seguinte conteúdo ao ficheiro de configuração **stunnel.conf** e, em seguida, guarde e saia.

```
debug = 4
output = /var/log/stunnel.log
sslVersion = all
[redis-client]
client = yes
accept = 8000
connect = {public access address}
CAfile = /etc/stunnel/dcs-ca.cer
```

Modifique os seguintes parâmetros conforme necessário e deixe os outros parâmetros inalterados:

- **client**: indica Stunnel. O valor fixo é **yes**.
- **CAfile**: especifica um certificado CA, que é opcional. Se um certificado CA for necessário, baixe e descomprima o certificado **dcs-ca.cer** conforme instruído em **Passo 2**. Se não for necessário, exclua este parâmetro.
- **accept**: especifica o número da porta de escuta definida pelo usuário do Stunnel. Especifique esse parâmetro ao acessar uma instância de DCS usando um cliente Redis.
- **connect**: especifica o endereço de encaminhamento e o número da porta do Stunnel. Defina esse parâmetro como o endereço de acesso público da instância obtido em **Passo 2**.

Segue-se um exemplo de configuração:

```
[redis-client]
client = yes
CAfile = D:\tmp\dcs\dcs-ca.cer
accept = 8000
connect = 49.**.**.211:36379
```

Passo 7 Execute os seguintes comandos para iniciar o Stunnel:

stunnel /*{customdir}*/stunnel.conf

No comando anterior, ***{customdir}*** indica o diretório de armazenamento personalizado para o arquivo **stunnel.conf** descrito em **Passo 5**. Segue-se um exemplo de comando:

stunnel /etc/stunnel/stunnel.conf

NOTA

Para o SO de Ubuntu, execute o comando **/etc/init.d/stunnel4 start** para iniciar o Stunnel. O nome do serviço ou processo é **stunnel4** para a versão Stunnel 4.x.

Depois de iniciar o cliente Stunnel, execute o comando **ps -ef|grep stunnel** para verificar se o processo está sendo executado corretamente.

Passo 8 Execute o seguinte comando para verificar se o Stunnel está sendo escutado:

netstat -plunt |grep *8000*|grep "LISTEN"

8000 indica o número da porta de escuta definida pelo usuário do Stunnel configurado no campo de **accept** em **Passo 6**.

Se uma linha contendo o número de porta **8000** for exibida no resultado retornado, o Stunnel está sendo executado corretamente. Quando o cliente do Redis se conecta ao endereço **127.0.0.1:8000**, o Stunnel encaminhará as solicitações para a instância do DCS Redis.

Passo 9 Acesse a instância do DCS Redis.

1. Efetue login no dispositivo Linux local.
2. Execute o seguinte comando para baixar o pacote de código-fonte do seu cliente Redis do **http://download.redis.io/releases/redis-5.0.8.tar.gz**:

   wget http://download.redis.io/releases/redis-5.0.8.tar.gz

   NOTA

   Você também pode instalar o cliente Redis executando o seguinte comando yum ou apt:
   - **yum install redis**
   - **apt install redis-server**
3. Execute o seguinte comando para descompactar o pacote de código-fonte do seu cliente Redis:

   tar -xzf redis-5.0.8.tar.gz
4. Execute os seguintes comandos para ir para o diretório do Redis e compilar o código-fonte do seu cliente Redis:

   cd redis-5.0.8

   make
5. Execute os seguintes comandos para acessar a instância do DCS Redis escolhida:

   cd src

./redis-cli -h 127.0.0.1 -p *8000*

CUIDADO

No comando anterior:

- O endereço a seguir **-h** indica o endereço do cliente Stunnel, que é **127.0.0.1**.
- A porta a seguir **-p** é a porta de escuta do cliente Stunnel, que foi configurada no campo **accept** em Passo 6. **8000** é usado como exemplo.

Não use o endereço de acesso público e a porta exibidos no console para os parâmetros **-h** e **-p**.

6. Digite a senha. Você pode ler e gravar dados em cache somente depois que a senha for verificada.

   auth *{password}*

   {password} indica a senha usada para efetuar login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

   Você acessou a instância com êxito se a seguinte saída do comando for exibida:

   ```
   OK
   127.0.0.1:8000>
   ```

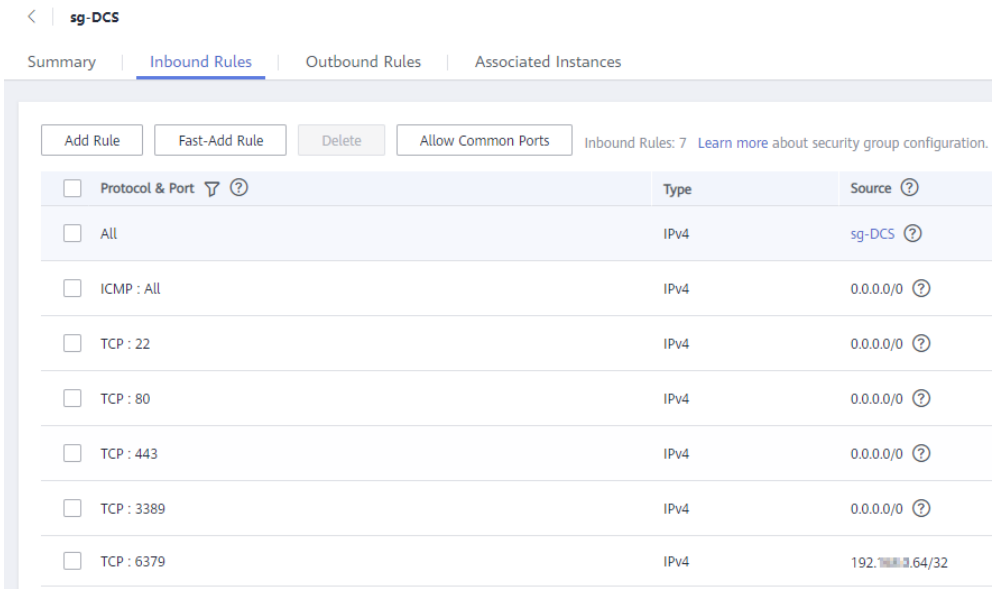
----Fim

Conectando-se ao Redis sem criptografia SSL

Passo 1 Assegure-se de que a regra do grupo de segurança permita o acesso público através da porta 6379.

Quando a criptografia SSL é desabilitada, o endereço de acesso público da instância pode ser acessado somente se o acesso pela porta 6379 for permitido.

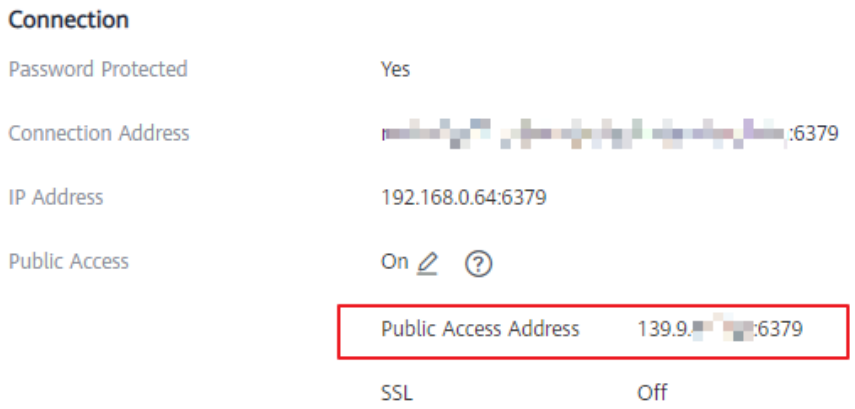
Figura 4-8 Regra de grupo de segurança (porta 6379)



Passo 2 Obtenha o endereço de acesso público da instância.

O endereço de acesso público é exibido na seção **Connection** da página Informações **Basic Information**.

Figura 4-9 Exibindo o endereço de acesso público (SSL desativado; porta 6379)



Passo 3 Efetue login no dispositivo Linux local.

Passo 4 Execute o seguinte comando para baixar o pacote de código-fonte do seu cliente Redis do http://download.redis.io/releases/redis-5.0.8.tar.gz:

wget http://download.redis.io/releases/redis-5.0.8.tar.gz

NOTA

Você também pode instalar o cliente Redis executando o seguinte comando yum ou apt:

- **yum install redis**
- **apt install redis-server**

Passo 5 Execute o seguinte comando para descompactar o pacote de código-fonte do seu cliente Redis:

tar -xzf redis-5.0.8.tar.gz

Passo 6 Execute os seguintes comandos para ir para o diretório do Redis e compilar o código-fonte do seu cliente Redis:

cd redis-5.0.8

fazer

Passo 7 Execute os seguintes comandos para acessar a instância do DCS Redis escolhida:

cd src

./redis-cli -h *{public access address}* -p 6379

Substitua *{public access address}* pelo endereço obtido em **Passo 2**. Por exemplo:

./redis-cli -h 49..**.211 -p 6379**

Passo 8 Digite a senha. Você pode ler e gravar dados em cache somente depois que a senha for verificada.

auth *{password}*

{password} indica a senha usada para efetuar login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Você acessou a instância com êxito se a seguinte saída do comando for exibida:

```
OK
49.**.**.211:6379>
```

----Fim

Solução de problemas

- Sintoma: "Erro: A redefinição da conexão por peer é exibida.

  Possible cause: O grupo de segurança está configurado incorretamente. Você precisa habilitar a porta **36379** ou **6379**.

- Quando o redis-cli é usado para se conectar a uma instância, a seguinte mensagem é exibida indicando que o host remoto fecha forçosamente uma conexão existente.

  Possible cause: A criptografia SSL foi habilitada, mas o Stunnel não está configurado durante a conexão. O endereço IP exibido no console foi usado para conexão. Neste caso, siga rigorosamente as instruções fornecidas em **Conectando-se ao Redis com criptografia SSL**..

- Para obter mais informações sobre falhas de conexão do Redis, consulte **Solução de problemas de Exceções de Conexão do Redis**.

4.3 Acesso em diferentes idiomas

4.3.1 redis-cli

Esta seção descreve como usar o redis-cli em um ECS na mesma VPC que uma instância do DCS Redis para se conectar à instância. Para obter detalhes sobre mais clientes, consulte o **site oficial do Redis**.

Para obter mais informações sobre como acessar uma instância do DCS Redis em redes públicas, consulte **Passo 3: Acessar uma instância do DCS Redis no Windows**.

NOTA

- O Redis 3.0 não suporta personalização de portas e permite apenas a porta 6379. Para Redis 4.0 e 5.0, você pode especificar uma porta ou usar a porta padrão 6379. O seguinte usa a porta padrão 6379. Se você especificou uma porta, substitua 6379 pela porta real.
- **When connecting to a Redis Cluster instance, ensure that -c is added to the command.** Caso contrário, a conexão falhará.
  - Execute o seguinte comando para conectar-se a uma instância do Cluster do Redis:

    ./redis-cli -h *{dcs_instance_address}* **-p 6379 -a** *{password}* **-c**
  - Execute o seguinte comando para se conectar a uma instância de cluster de proxy, principal/em espera ou de nó único:

    ./redis-cli -h *{dcs_instance_address}* **-p 6379 -a** *{password}*

  Para mais detalhes, consulte **Passo 3**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, verifique se o ambiente de compilação GCC foi instalado no ECS.

Procedimento (Linux)

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para mais detalhes, consulte **Exibindo Detalhes da Instância**.

Passo 2 Para obter detalhes sobre como instalar o cliente redis-cli, consulte as **instruções oficiais do Redis**.

As etapas a seguir pressupõem que seu cliente esteja instalado no SO de Linux.

1. Acesse o ECS.
2. Execute o seguinte comando para baixar o pacote de código-fonte do seu cliente Redis do **http://download.redis.io/releases/redis-5.0.8.tar.gz**:

   wget http://download.redis.io/releases/redis-5.0.8.tar.gz
3. Execute o seguinte comando para descompactar o pacote de código-fonte do seu cliente Redis:

   tar -xzf redis-5.0.8.tar.gz
4. Execute os seguintes comandos para ir para o diretório do Redis e compilar o código-fonte do seu cliente Redis:

   cd redis-5.0.8

   make

   cd src

Passo 3 Acesse a instância do DCS Redis.

- Acessar uma instância de DCS de um tipo diferente do Cluster do Redis.

  Execute o procedimento a seguir para acessar uma instância do DCS Redis 3.0 ou uma instância do DCS Redis 4.0 ou 5.0 de cluster de nó único, principal/em espera ou proxy.

  a. Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

  ./redis-cli -h** {dcs_instance_address} **-p 6379

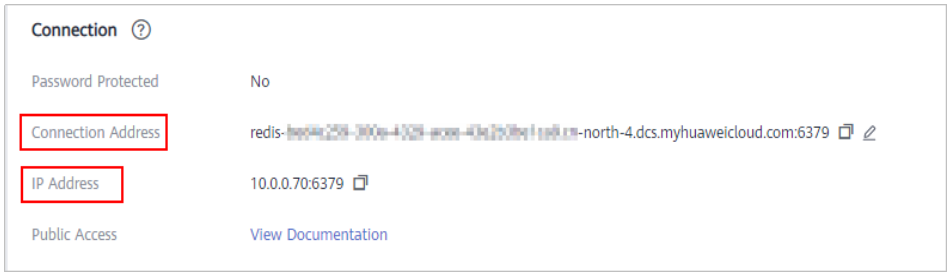
  {dcs_instance_address} indica o endereço IP/nome de domínio da instância DCS e **6379** é a porta usada para acessar a instância. O endereço IP/nome do domínio e o número da porta são obtidos em **Passo 1**.

  NOTA

  Para uma instância do Cluster de Proxy DCS Redis, você pode usar **Connection Address** ou **IP Address** para *{dcs_instance_address}*. Os endereços podem ser obtidos na página de informações básicas da instância no console, conforme mostrado na **Figura 4-10**.

  - **Connection Address** e **IP Address** são os endereços LB. As solicitações são distribuídas pelos nós proxy.
  - Você pode usar **Backend Addresses** para se conectar diretamente ao nó proxy especificado de uma instância do DCS Redis 3.0 de cluster de proxy.

Figura 4-10 Obtendo os endereços para conexão a instâncias de DCS do Cluster de Proxy



O exemplo a seguir usa o endereço de nome de domínio de uma instância do DCS Redis. Altere o nome de domínio e a porta conforme necessário.

```
[root@ecs-redis redis-5.0.8]# cd src
[root@ecs-redis src]# ./redis-cli -h redis-069949a-dcs-
lxy.dcs.huaweicloud.com -p 6379
redis-069949a-dcs-lxy.dcs.huaweicloud.com:6379>
```

b. Se você definiu uma senha para a instância do DCS, digite a senha nesta etapa. Você pode ler e gravar dados em cache somente depois que a senha for verificada.

auth *{password}*

{password} indica a senha usada para efetuar login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

A saída do comando é a seguinte:

```
redis-069949a-dcs-lxy.dcs.huaweicloud.com:6379> auth *******
OK
redis-069949a-dcs-lxy.dcs.huaweicloud.com:6379>
```

- Acesse uma instância de DCS do tipo Cluster do Redis.

  Execute o procedimento a seguir para acessar uma instância do DCS Redis 4.0 ou 5.0 no tipo de cluster do Redis.

  a. Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

  ./redis-cli -h *{dcs_instance_address}* -p 6379 -a *{password}* -c

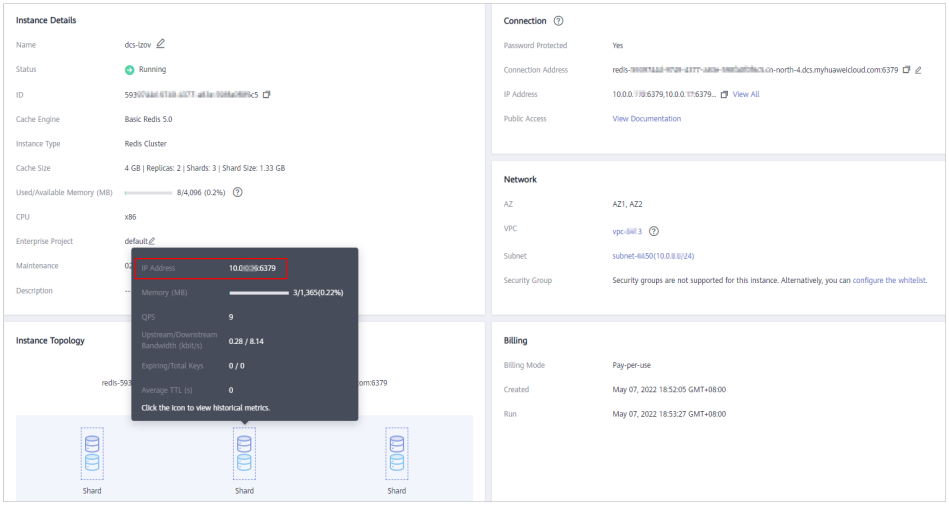
  {dcs_instance_address} indica o endereço IP/nome de domínio da instância do DCS Redis, **6379** é a porta usada para acessar a instância, *{password}* é a senha da instância, e **-c** é usado para acessar nós do Cluster do Redis. O endereço IP/nome do domínio e o número da porta são obtidos em **Passo 1**.

  NOTA

  Você pode definir *{dcs_instance_address}* como **Connection Address** ou **IP Address** na seção **Connection** ou **IP Address** na seção **Instance Topology**. Os endereços podem ser obtidos na página de informações básicas da instância no console, conforme mostrado na **Figura 4-11**.

  - O campo **IP Address** fornece dois endereços IP. Você pode usar qualquer um deles para se conectar à instância. O algoritmo CRC16 (chave) mod 16384 é usado para calcular o que é o slot de hash de uma determinada chave.
  - Usando o **IP Address** na seção **Instance Topology**, você pode se conectar ao estilhaço especificado.

Figura 4-11 Obtendo os endereços para conexão com uma instância de DCS do Cluster do Redis



- O exemplo a seguir usa o endereço IP de uma instância do DCS Redis. Altere o endereço IP e a porta conforme necessário.

  ```
  root@ecs-redis:~/redis-5.0.8/src# ./redis-cli -h 192.168.0.85 -p
  6379 -a ****** -c
  192.168.0.85:6379>
  ```

- O exemplo a seguir usa o nome de domínio de uma instância do DCS Redis. Altere o nome de domínio e a porta conforme necessário.

  ```
  root@ecs-redis:~/redis-5.0.8/src# ./redis-cli -h redis-51e463c-dcs-
  lxy.dcs.huaweicloud.com -p 6379 -a ****** -c
  redis-51e463c-dcs-lxy.dcs.huaweicloud.com:6379>
  ```

b. Execute o seguinte comando para exibir as informações de nó do Cluster do Redis:

cluster nodes

Cada estilhaço em um cluster do Redis tem um principal e uma réplica por padrão. O comando procedente fornece todas as informações dos nós do cluster.

```
192.168.0.85:6379> cluster nodes
0988ae8fd3686074c9afdcce73d7878c81a33ddc 192.168.0.231:6379@16379 slave
f0141816260ca5029c56333095f015c7a058f113 0 1568084030
000 3 connected
1a32d809c0b743bd83b5e1c277d5d201d0140b75 192.168.0.85:6379@16379
myself,master - 0 1568084030000 2 connected 5461-10922
c8ad7af9a12cce3c8e416fb67bd6ec9207f0082d 192.168.0.130:6379@16379 slave
1a32d809c0b743bd83b5e1c277d5d201d0140b75 0 1568084031
000 2 connected
7ca218299c254b5da939f8e60a940ac8171adc27 192.168.0.22:6379@16379 master
- 0 1568084030000 1 connected 0-5460
f0141816260ca5029c56333095f015c7a058f113 192.168.0.170:6379@16379 master
- 0 1568084031992 3 connected 10923-16383
19b1a400815396c6223963b013ec934a657bdc52 192.168.0.161:6379@16379 slave
7ca218299c254b5da939f8e60a940ac8171adc27 0 1568084031
000 1 connected
```

As operações de gravação só podem ser executadas em nós principais. O algoritmo CRC16 (chave) mod 16384 é usado para calcular qual é o slot de hash de uma determinada chave.

Conforme mostrado a seguir, o valor do **CRC16 (KEY) mode 16384** determina o hash slot em que uma determinada chave está localizada e redireciona o cliente para o nó em que o hash slot está localizado.

```
192.168.0.170:6379> set hello world
-> Redirected to slot [866] located at 192.168.0.22:6379
```



```
OK
192.168.0.22:6379> set happy day
OK
192.168.0.22:6379> set abc 123
-> Redirected to slot [7638] located at 192.168.0.85:6379
OK
192.168.0.85:6379> get hello
-> Redirected to slot [866] located at 192.168.0.22:6379
"world"
192.168.0.22:6379> get abc
-> Redirected to slot [7638] located at 192.168.0.85:6379
"123"
192.168.0.85:6379>
```

----Fim

Procedimento (Windows)

Download o pacote de instalação do cliente Windows Redis. Descompacte o pacote, abra a ferramenta CLI **cmd.exe** e vá para o diretório onde o pacote de instalação do cliente Redis descompactado foi salvo. Em seguida, execute o seguinte comando para acessar a instância do DCS Redis:

redis-cli -h *XXX* -p *6379*

XXX indica o endereço IP/nome de domínio da instância DCS e **6379** é um número de porta de exemplo usado para acessar a instância DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Exibindo Detalhes da Instância**. Altere o endereço IP/nome do domínio e a porta conforme necessário.

4.3.2 Java

4.3.2.1 Jedis

Acesse uma instância do DCS Redis por meio de Jedis em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, certifique-se de que o ambiente de compilação Java tenha sido instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Passo 3 Use o Maven para adicionar a seguinte dependência ao arquivo **pom.xml**:


```
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>4.1.1</version>
</dependency>
```

Passo 4 Acesse a instância do DCS usando Jedis.

Obtenha o **código fonte** do cliente Jedi. Use um dos dois métodos a seguir para acessar uma instância do DCS Redis por meio de Jedis:

- Conexão de Jedis Únicos
- Piscina Jedis

Exemplo de código:

1. Exemplo de uso de Jedis para se conectar a uma instância DCS Redis de cluster de nó único, mestre/em espera ou proxy com uma única conexão

```
//Creating a connection in password mode
 String host = "192.168.0.150";
 int port = 6379;
 String pwd = "passwd";

 Jedis client = new Jedis(host, port);
 client.auth(pwd);
 client.connect();
//Run the SET command.
 String result = client.set("key-string", "Hello, Redis!");
 System.out.println( String.format("set command result:%s", result) );
//Run the GET command.
 String value = client.get("key-string");
System.out.println( String.format("get command result:%s", value) );

//Creating a connection in password-free mode
 String host = "192.168.0.150";
 int port = 6379;

 Jedis client = new Jedis(host, port);
 client.connect();
//Run the SET command.
 String result = client.set("key-string", "Hello, Redis!");
 System.out.println( String.format("set command result:%s", result) );
//Run the GET command.
 String value = client.get("key-string");
 System.out.println( String.format("get command result:%s", value) );
```

host indica o exemplo de endereço IP/nome de domínio da instância do DCS e a *port* indica o número da porta da instância do DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/domínio e a porta conforme necessário. *pwd* indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

2. Exemplo de uso de Jedis para conexão a uma instância de cluster DCS Redis de nó único, principal/em espera ou proxy com pool de conexão

```
//Generate configuration information of a Jedis pool
 String ip = "192.168.0.150";
 int port = 6379;
 String pwd = "passwd";
 GenericObjectPoolConfig config = new GenericObjectPoolConfig();
 config.setTestOnBorrow(false);
 config.setTestOnReturn(false);
 config.setMaxTotal(100);
 config.setMaxIdle(100);
 config.setMaxWaitMillis(2000);
JedisPool pool = new JedisPool(config, ip, port, 100000, pwd);//Generate a
Jedis pool when the application is being initialized
```



```
//Get a Jedis connection from the Jedis pool when a service operation occurs
 Jedis client = pool.getResource();
 try {
     //Run commands
     String result = client.set("key-string", "Hello, Redis!");
     System.out.println( String.format("set command result:%s", result) );
     String value = client.get("key-string");
     System.out.println( String.format("get command result:%s", value) );
 } catch (Exception e) {
     // TODO: handle exception
 } finally {
     //Return the Jedis connection to the Jedis pool when the service
operation is completed
     if (null != client) {
         pool.returnResource(client);
     }
 } // end of try block
 //Destroy the Jedis pool when the application is closed
 pool.destroy();

//Configure the connection pool in password-free mode
 String ip = "192.168.0.150";
 int port = 6379;
 GenericObjectPoolConfig config = new GenericObjectPoolConfig();
 config.setTestOnBorrow(false);
 config.setTestOnReturn(false);
 config.setMaxTotal(100);
 config.setMaxIdle(100);
 config.setMaxWaitMillis(2000);
 JedisPool pool = new JedisPool(config, ip, port, 100000);//Generate a
JedisPool when the application is being initialized
 //Get a Jedis connection from the Jedis pool when a service operation occurs
 Jedis client = pool.getResource();
 try {
     //Run commands
     String result = client.set("key-string", "Hello, Redis!");
     System.out.println( String.format("set command result:%s", result) );
     String value = client.get("key-string");
     System.out.println( String.format("get command result:%s", value) );
 } catch (Exception e) {
     // TODO: handle exception
 } finally {
     //Return the Jedis connection to the Jedis pool when the service
operation is completed
     if (null != client) {
         pool.returnResource(client);
     }
 } // end of try block
 //Destroy the Jedis pool when the application is closed
 pool.destroy();
```

ip indica o endereço IP/nome de domínio da instância DCS e *port* indica o número da porta da instância DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço de IP/domínio e a porta conforme necessário. *pwd* indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

3. Exemplo de código para conexão com o Cluster do Redis usando uma única conexão

   – Com uma senha

   ```
   //The following shows password-protected access.
   int port = 6379;
   String host = "192.168.144.37";
   //Create JedisCluster.
   Set<HostAndPort> nodes = new HashSet<HostAndPort>();
   nodes.add(new HostAndPort(host, port));
   JedisCluster cluster = new JedisCluster(nodes, 5000, 3000, 10,
   "password", new JedisPoolConfig());
   cluster.set("key", "value");
   ```



```
System.out.println("Connected to RedisCluster:" + cluster.get("key"));
cluster.close();
```

– Sem uma senha

```
int port = 6379;
String host = "192.168.144.37";
//Create JedisCluster.
Set<HostAndPort> nodes = new HashSet<HostAndPort>();
nodes.add(new HostAndPort(host, port));
JedisCluster cluster = new JedisCluster(nodes);
cluster.set("key", "value");
System.out.println("Connected to RedisCluster:" + cluster.get("key"));
cluster.close();
```

host indica o exemplo de endereço IP/nome de domínio da instância do DCS e a *port* indica o número da porta da instância do DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/domínio e a porta conforme necessário. *password* indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Passo 5 Compile o código de acordo com o arquivo **readme** no código fonte do cliente Jedis. Execute o cliente Jedis para acessar a instância do DCS Redis escolhida.

----Fim

4.3.2.2 Alface

Acesse uma instância do DCS Redis por meio do Lettuce em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, certifique-se de que o ambiente de compilação Java tenha sido instalado no ECS.

Procedimento

Passo 1 Exiba o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Passo 3 Use o Maven para adicionar a seguinte dependência ao arquivo **pom.xml**:

```
<dependency>
  <groupId>io.lettuce</groupId>
  <artifactId>lettuce-core</artifactId>
  <version>6.1.6.RELEASE</version>
</dependency>
```

Passo 4 Use Lettuce (um cliente Java) para se conectar à instância do DCS.

- Exemplo de uso do Lettuce para conexão a uma instância de Cluster DCS Redis de nó único, mestre/em espera ou proxy com uma única conexão

```
// password indicates the connection password. If there is no password,
delete "password@". If there is a password and it contains special
characters, conversion is required.
RedisClient redisClient = RedisClient.create("redis://password@host:port");
StatefulRedisConnection<String, String> connection = redisClient.connect();
RedisCommands<String, String> syncCommands = connection.sync();
syncCommands.set("key", "value");
System.out.println("Connected to Redis:" + syncCommands.get("key"));
// Close the connection.
connection.close();
// Close the client.
redisClient.shutdown();
```

- Exemplo de uso do Lettuce para conectar-se a uma instância do DCS Redis de cluster de nó único, mestre/em espera ou proxy com pool de conexão

  a. Adicione a seguinte dependência além da dependência Maven anterior:

  ```
  <dependency>
     <groupId>org.apache.commons</groupId>
     <artifactId>commons-pool2</artifactId>
      <version>2.11.1</version>
   </dependency>
  ```

  b. O código é o seguinte:

  ```
  // password indicates the connection password. If there is no password,
  delete "password@". If there is a password and it contains special
  characters, conversion is required.
  RedisClient clusterClient = RedisClient.create("redis://
  password@host:port");
  GenericObjectPoolConfig<StatefulRedisConnection<String, String>>
  genericObjectPoolConfig = new GenericObjectPoolConfig();
  // Connection pool parameters
  genericObjectPoolConfig.setMaxIdle(3);
  genericObjectPoolConfig.setMinIdle(2);
  genericObjectPoolConfig.setMaxTotal(3);
  genericObjectPoolConfig.setMaxWaitMillis(-1);
  GenericObjectPool<StatefulRedisConnection<String, String>> pool =
  ConnectionPoolSupport
          .createGenericObjectPool(() -> clusterClient.connect(),
  genericObjectPoolConfig);
  // Obtain a connection to perform operations.
  try (StatefulRedisConnection<String, String> con = pool.borrowObject()) {
      RedisCommands<String, String> sync = con.sync();
      sync.set("key", "value");
      System.out.println("Connected by pool:" + sync.get("key"));
  } catch (Exception e) {
      e.printStackTrace();
  }finally {
      // Close the resources.
      pool.close();
      clusterClient.shutdown();
  }
  ```

- Exemplo de uso do Lettuce para se conectar a uma instância do Cluster do Redis DCS do Redis com uma única conexão (a atualização automatizada da topologia deve estar ativada)

  ```
  public class SingleConnectionToCluster {
      public static void main(String[] args) {
           // Enable automated topology refresh.
          ClusterTopologyRefreshOptions topologyRefreshOptions =
  ClusterTopologyRefreshOptions.builder()
           // Periodic refresh: every time milliseconds.
              .enablePeriodicRefresh(Duration.ofMillis(time))
           // Triggers of adaptive refresh: MOVED redirection, ASK redirection,
  reconnection, unknown node (since 5.1), and slot not in any of the current
  shards (since 5.2).
              .enableAllAdaptiveRefreshTriggers()
              .build();
          // password indicates the connection password. If there is no
  password, delete "password@". If there is a password and it contains special
  ```



```
characters, conversion is required.
        RedisClusterClient redisClient = RedisClusterClient.create("redis://
password@host:port");
        redisClient.setOptions(ClusterClientOptions.builder()
            .topologyRefreshOptions(topologyRefreshOptions)
            .build());
        StatefulRedisClusterConnection<String, String> connection =
redisClient.connect();
         // Preferentially read data from the replicas.
         connection.setReadFrom(ReadFrom.REPLICA_PREFERRED);
        RedisAdvancedClusterCommands<String, String> syncCommands =
connection.sync();
        syncCommands.set("key", "value");
        System.out.println("Connected to RedisCluster:" +
syncCommands.get("key"));
        // Close the connection.
        connection.close();
        // Close the client.
        redisClient.shutdown();
    }
}
```

- Exemplo de código para conexão ao Cluster do Redis com pool de conexão

  a. Adicione a seguinte dependência além da dependência Maven anterior:

  ```
  <dependency>
     <groupId>org.apache.commons</groupId>
     <artifactId>commons-pool2</artifactId>
      <version>2.11.1</version>
   </dependency>
  ```

  b. O código é o seguinte (a atualização automatizada da topologia deve estar ativada):

  ```
  public class PoolConnectionToCluster {
      public static void main(String[] args) {
           // Enable automated topology refresh.
          ClusterTopologyRefreshOptions topologyRefreshOptions =
  ClusterTopologyRefreshOptions.builder()
           // Periodic refresh every time milliseconds.
              .enablePeriodicRefresh(Duration.ofMillis(time))
           // Triggers of adaptive refresh: MOVED redirection, ASK
  redirection, reconnection, unknown node (since 5.1), and slot not in any
  of the current shards (since 5.2).
              .enableAllAdaptiveRefreshTriggers()
              .build();
          // password indicates the connection password. If there is no
  password, delete "password@". If there is a password and it contains
  special characters, conversion is required.
          RedisClusterClient redisClient =
  RedisClusterClient.create("redis://password@host:port");
          redisClient.setOptions(ClusterClientOptions.builder()
              .topologyRefreshOptions(topologyRefreshOptions)
              .build());
          GenericObjectPoolConfig<StatefulRedisClusterConnection<String,
  String>> genericObjectPoolConfig
              = new GenericObjectPoolConfig();
          // Connection pool parameters
          genericObjectPoolConfig.setMaxIdle(3);
          genericObjectPoolConfig.setMinIdle(2);
          genericObjectPoolConfig.setMaxTotal(3);

  genericObjectPoolConfig.setTimeBetweenEvictionRuns(Duration.ofMillis(2000
  ));
          genericObjectPoolConfig.setMaxWait(Duration.ofMillis(5000));
          GenericObjectPool<StatefulRedisClusterConnection<String,
  String>> pool = ConnectionPoolSupport
              .createGenericObjectPool(() -> redisClient.connect(),
  genericObjectPoolConfig);
          // Obtain a connection to perform operations.
          try (StatefulRedisClusterConnection<String, String> con =
  pool.borrowObject()) {
              // Preferentially read data from the replicas.
  ```



```
            con.setReadFrom(ReadFrom.REPLICA_PREFERRED);
            RedisAdvancedClusterCommands<String, String> syncCommands =
con.sync();
            syncCommands.set("key", "value");
            System.out.println("Connected to RedisCluster:" +
syncCommands.get("key"));
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
    // Close the resources.
            pool.close();
            redisClient.shutdown();
        }
    }
}
```

host é o endereço IP/nome de domínio da instância do DCS, **port** é o número da porta da instância do DCS e **password** é a senha da instância do DCS. Especifique esses parâmetros conforme necessário antes de executar o código. O pool de conexões é recomendado. Ajuste parâmetros como **timeout**, **MaxTotal** (número máximo de conexões), **MinIdle** (número mínimo de conexões ociosas), **MaxIdle** (número máximo de conexões ociosas) e **MaxWait** (tempo máximo de espera) com base nos requisitos de serviço.

----Fim

4.3.2.3 Redisson

Acesse uma instância do DCS Redis por meio do Redisson em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

NOTA

- Se uma senha tiver sido definida durante a criação da instância do DCS Redis, configure a senha para conexão com o Redis usando o Redisson. Não codifique duramente a senha de texto simples.
- Para conectar a uma instância de cluster de proxy, mestre/em espera ou de nó único, use o método **useSingleServer** do objeto **SingleServerConfig** do Redisson. Para se conectar a uma instância do Cluster do Redis, use o método **useClusterServers** do objeto **ClusterServersConfig**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, certifique-se de que o ambiente de compilação Java tenha sido instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Passo 3 Use o Maven para adicionar a seguinte dependência ao arquivo **pom.xml**:

```
<dependency>
  <groupId>org.redisson</groupId>
```



```
  <artifactId>redisson</artifactId>
  <version>3.16.8</version>
</dependency>
```

Passo 4 Acesse a instância do DCS usando o Redisson (um cliente Java).

- Exemplo de uso do Redisson para se conectar a uma instância do DCS Redis de cluster de nó único, mestre/em espera ou proxy com uma única conexão

```
Config config = new Config();
SingleServerConfig singleServerConfig = config.useSingleServer();
singleServerConfig.setAddress("redis://host:port");
// singleServerConfig.setPassword("9client!");
RedissonClient redisson = Redisson.create(config);
//Test concurrentMap. Data is synchronized to Redis when the put method is
used.
ConcurrentMap<String, Object> map = redisson.getMap("FirstMap");
map.put("wanger", "male");
map.put("zhangsan", "nan");
map.put("lisi", "female");
ConcurrentMap resultMap = redisson.getMap("FirstMap");
System.out.println("resultMap==" + resultMap.keySet());
//Test Set
Set mySet = redisson.getSet("MySet");
mySet.add("wanger");
mySet.add("lisi");
Set resultSet = redisson.getSet("MySet");
System.out.println("resultSet===" + resultSet.size());
//Test Queue
Queue myQueue = redisson.getQueue("FirstQueue");
myQueue.add("wanger");
myQueue.add("lili");
myQueue.add("zhangsan");
myQueue.peek();
myQueue.poll();
Queue resultQueue = redisson.getQueue("FirstQueue");
System.out.println("resultQueue===" + resultQueue);
//Close the connection.
redisson.shutdown();
```

- Exemplo de uso do Redisson para conectar-se a uma instância de Cluster DCS Redis de nó único, mestre/em espera ou proxy com pool de conexão

```
//1. Initialization
Config config = new Config();
SingleServerConfig singleServerConfig = config.useSingleServer();
singleServerConfig.setAddress("redis://host:6379");
//Set the maximum number of connections in the connection pool of the master
node to 500.
singleServerConfig.setConnectionPoolSize(500);
//The connections will be automatically closed and removed from the
connection pool. The time unit is millisecond.
singleServerConfig.setIdleConnectionTimeout(10000);
RedissonClient redisson = Redisson.create(config);
//Test concurrentMap. Data is synchronized to Redis when the put method is
used.
ConcurrentMap<String, Object> map = redisson.getMap("FirstMap");
map.put("wanger", "male");
map.put("zhangsan", "nan");
map.put("lisi", "female");
ConcurrentMap resultMap = redisson.getMap("FirstMap");
System.out.println("resultMap==" + resultMap.keySet());
//Test Set
Set mySet = redisson.getSet("MySet");
mySet.add("wanger");
mySet.add("lisi");
Set resultSet = redisson.getSet("MySet");
System.out.println("resultSet===" + resultSet.size());
//Test Queue
Queue myQueue = redisson.getQueue("FirstQueue");
myQueue.add("wanger");
myQueue.add("lili");
```



```
myQueue.add("zhangsan");
myQueue.peek();
myQueue.poll();
Queue resultQueue = redisson.getQueue("FirstQueue");
System.out.println("resultQueue===" + resultQueue);
//Close the connection.
redisson.shutdown();
```

- Exemplo de uso do Redisson para se conectar a um cluster do Redis

```
Config config = new Config();
ClusterServersConfig clusterServersConfig = config.useClusterServers();
clusterServersConfig.addNodeAddress("redis://host:port");
//Set a password.
// clusterServersConfig.setPassword("");
RedissonClient redisson = Redisson.create(config);
ConcurrentMap<String, Object> map = redisson.getMap("FirstMap");
map.put("wanger", "male");
map.put("zhangsan", "nan");
map.put("lisi", "female");
ConcurrentMap resultMap = redisson.getMap("FirstMap");
System.out.println("resultMap==" + resultMap.keySet());
//2. Test Set
Set mySet = redisson.getSet("MySet");
mySet.add("wanger");
mySet.add("lisi");
Set resultSet = redisson.getSet("MySet");
System.out.println("resultSet===" + resultSet.size());
//3. Test Queue
Queue myQueue = redisson.getQueue("FirstQueue");
myQueue.add("wanger");
myQueue.add("lili");
myQueue.add("zhangsan");
myQueue.peek();
myQueue.poll();
Queue resultQueue = redisson.getQueue("FirstQueue");
System.out.println("resultQueue===" + resultQueue);
//Close the connection.
redisson.shutdown();
```

----Fim

4.3.3 Integração de alface com Spring Boot

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, certifique-se de que o ambiente de compilação Java tenha sido instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para mais detalhes, consulte **Exibindo Detalhes da Instância**.

Passo 2 Acesse o ECS.

Passo 3 Use o Maven para adicionar a seguinte dependência ao arquivo **pom.xml**:

NOTA

- Desde o Spring Boot 2.0, o Lettuce é usado como cliente padrão para conexões.
- Spring Boot 2.6.6 e Lettuce 6.1.8 são usados.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

Passo 4 Use o Spring Boot integrado ao Lettuce para se conectar à instância.

- Exemplo de uso do Spring Boot e do Lettuce para conectar-se a uma instância do DCS Redis de cluster de nó único, mestre/standby ou proxy com uma única conexão

  a. Adicione a configuração do Redis ao arquivo de configuração **application.properties**.

  ```
  spring.redis.host=host
  spring.redis.database=0
  spring.redis.password=pwd
  spring.redis.port=port
  ```

  b. Classe de configuração do Redis RedisConfiguration

  ```
  @Bean
  public RedisTemplate<String, Object>
  redisTemplate(LettuceConnectionFactory lettuceConnectionFactory) {
      RedisTemplate<String, Object> template = new RedisTemplate<>();
      template.setConnectionFactory(lettuceConnectionFactory);
        // Replace the default JdkSerializationRedisSerializer with
  Jackson2JsonRedisSerializer to serialize and deserialize the Redis value.
      Jackson2JsonRedisSerializer<Object> jackson2JsonRedisSerializer =
  new Jackson2JsonRedisSerializer<>(Object.class);
      ObjectMapper mapper = new ObjectMapper();
      mapper.setVisibility(PropertyAccessor.ALL,
  JsonAutoDetect.Visibility.ANY);
      mapper.activateDefaultTyping(LaissezFaireSubTypeValidator.instance,
          ObjectMapper.DefaultTyping.NON_FINAL,
  JsonTypeInfo.As.PROPERTY);
      jackson2JsonRedisSerializer.setObjectMapper(mapper);
      StringRedisSerializer stringRedisSerializer = new
  StringRedisSerializer();
        // String serialization of keys
      template.setKeySerializer(stringRedisSerializer);
        // String serialization of hash keys
      template.setHashKeySerializer(stringRedisSerializer);
        // Jackson serialization of values
      template.setValueSerializer(jackson2JsonRedisSerializer);
        // Jackson serialization of hash values
      template.setHashValueSerializer(jackson2JsonRedisSerializer);
      template.afterPropertiesSet();
      return template;
  }
  ```

  c. Classe de operação do Redis RedisUtil

  ```
   /**
    * Obtain data from the cache.
    * @param key
    * @return value
    */
   public Object get(String key){
       return key==null?null:redisTemplate.opsForValue().get(key);
   }

   /**
    * Write data to the cache.
  ```



```
  * @param key
  * @param value
  * @return true (successful) false (failed)
  */
 public boolean set(String key,Object value) {
     try {
         redisTemplate.opsForValue().set(key, value);
         return true;
     } catch (Exception e) {
         e.printStackTrace();
         return false;
     }
 }
```

d. Escreva a classe do controlador para testes.

```
  @RestController
public class HelloRedis {
    @Autowired
    RedisUtil redisUtil;


    @RequestMapping("/setParams")
    @ResponseBody
    public String setParams(String name) {
            redisUtil.set("name", name);
            return "success";
        }

    @RequestMapping("/getParams")
    @ResponseBody
    public String getParams(String name) {
    System.out.println("--------------" + name + "-------------");
    String retName = redisUtil.get(name) + "";
    return retName;
    }

    }
```

- Exemplo de uso do Spring Boot e do Lettuce para conectar-se a uma instância do DCS Redis de cluster de nó único, mestre/standby ou proxy com pool de conexão

  a. Adicione a seguinte dependência além da dependência Maven anterior:

```
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-pool2</artifactId>
</dependency>
```

  b. Adicione a configuração do Redis ao arquivo de configuração **application.properties**.

```
spring.redis.host=host
spring.redis.database=0
spring.redis.password=pwd
spring.redis.port=port
# Connection timeout.
spring.redis.timeout=1000
# Maximum number of connections in the connection pool. A negative value
indicates no limit.
spring.redis.lettuce.pool.max-active=50
# Minimum number of idle connections in the connection pool.
spring.redis.lettuce.pool.min-idle=5
# Maximum number of idle connections in the connection pool.
spring.redis.lettuce.pool.max-idle=50
# Maximum time for waiting for connections in the connection pool. A
negative value indicates no limit.
spring.redis.lettuce.pool.max-wait=5000
# Interval for scheduling an eviction thread.
spring.redis.pool.time-between-eviction-runs-millis=2000
```

  c. Classe de configuração da conexão Redis RedisConfiguration

```
@Bean
public RedisTemplate<String, Object>
```



```
redisTemplate(LettuceConnectionFactory lettuceConnectionFactory) {
    lettuceConnectionFactory.setShareNativeConnection(false);
    RedisTemplate<String, Object> template = new RedisTemplate<>();
    template.setConnectionFactory(lettuceConnectionFactory);
    // Use Jackson2JsonRedisSerializer to replace the default
JdkSerializationRedisSerializer to serialize and deserialize the Redis
value.
    Jackson2JsonRedisSerializer<Object> jackson2JsonRedisSerializer =
new Jackson2JsonRedisSerializer<>(Object.class);
    ObjectMapper mapper = new ObjectMapper();
    mapper.setVisibility(PropertyAccessor.ALL,
JsonAutoDetect.Visibility.ANY);
    mapper.activateDefaultTyping(LaissezFaireSubTypeValidator.instance,
        ObjectMapper.DefaultTyping.NON_FINAL,
JsonTypeInfo.As.PROPERTY);
    jackson2JsonRedisSerializer.setObjectMapper(mapper);
    StringRedisSerializer stringRedisSerializer = new
StringRedisSerializer();
    // String serialization of keys
    template.setKeySerializer(stringRedisSerializer);
    // String serialization of hash keys
    template.setHashKeySerializer(stringRedisSerializer);
    // Jackson serialization of values
    template.setValueSerializer(jackson2JsonRedisSerializer);
    // Jackson serialization of hash values
    template.setHashValueSerializer(jackson2JsonRedisSerializer);
    template.afterPropertiesSet();
    return template;
}
```

- Exemplo de código para usar o Spring Boot e o Lettuce para se conectar ao Redis Cluster usando uma única conexão

  a. Adicione a configuração do Redis ao arquivo de configuração **application.properties**.

  ```
  spring.redis.cluster.nodes=host:port
  spring.redis.cluster.max-redirects=3
  spring.redis.password= pwd
  # Automated refresh interval
  spring.redis.lettuce.cluster.refresh.period=60
  # Enable automated refresh
  spring.redis.lettuce.cluster.refresh.adaptive=true
  spring.redis.timeout=60
  ```

  b. Classe de configuração RedisConfiguration do Redis (a atualização de topologia automatizada deve estar habilitada).

  ```
  @Bean
  public LettuceConnectionFactory lettuceConnectionFactory() {
       String[] nodes = clusterNodes.split(",");
       List<RedisNode> listNodes = new ArrayList();
       for (String node : nodes) {
           String[] ipAndPort = node.split(":");
           RedisNode redisNode = new RedisNode(ipAndPort[0],
  Integer.parseInt(ipAndPort[1]));
           listNodes.add(redisNode);
       }
       RedisClusterConfiguration redisClusterConfiguration = new
  RedisClusterConfiguration();
       redisClusterConfiguration.setClusterNodes(listNodes);
       redisClusterConfiguration.setPassword(password);
       redisClusterConfiguration.setMaxRedirects(maxRedirects);
        // Configure automated topology refresh.
       ClusterTopologyRefreshOptions topologyRefreshOptions =
  ClusterTopologyRefreshOptions.builder()
           .enablePeriodicRefresh(Duration.ofSeconds(period)) // Refresh
  the topology periodically.
           .enableAllAdaptiveRefreshTriggers() // Refresh the topology
  based on events.
           .build();
  ```



```
     ClusterClientOptions clusterClientOptions =
ClusterClientOptions.builder()
         // Redis command execution timeout. Only when the command
execution times out will a reconnection be triggered using the new
topology.
         .timeoutOptions(TimeoutOptions.enabled(Duration.ofSeconds(period
)))
         .topologyRefreshOptions(topologyRefreshOptions)
         .build();
     LettuceClientConfiguration clientConfig =
LettucePoolingClientConfiguration.builder()
             .commandTimeout(Duration.ofSeconds(timeout))
             .readFrom(ReadFrom.REPLICA_PREFERRED) // Preferentially
read data from the replicas.
             .clientOptions(clusterClientOptions)
             .build();
     LettuceConnectionFactory factory = new
LettuceConnectionFactory(redisClusterConfiguration, clientConfig);
     return factory;
}

@Bean
public RedisTemplate<String, Object>
redisTemplate(LettuceConnectionFactory lettuceConnectionFactory) {
    RedisTemplate<String, Object> template = new RedisTemplate<>();
    template.setConnectionFactory(lettuceConnectionFactory);
     // Use Jackson2JsonRedisSerializer to replace the default
JdkSerializationRedisSerializer to serialize and deserialize the Redis
value.
    Jackson2JsonRedisSerializer<Object> jackson2JsonRedisSerializer =
new Jackson2JsonRedisSerializer<>(Object.class);
    ObjectMapper mapper = new ObjectMapper();
    mapper.setVisibility(PropertyAccessor.ALL,
JsonAutoDetect.Visibility.ANY);
    mapper.activateDefaultTyping(LaissezFaireSubTypeValidator.instance,
        ObjectMapper.DefaultTyping.NON_FINAL,
JsonTypeInfo.As.PROPERTY);
    jackson2JsonRedisSerializer.setObjectMapper(mapper);
    StringRedisSerializer stringRedisSerializer = new
StringRedisSerializer();
     // String serialization of keys
    template.setKeySerializer(stringRedisSerializer);
     // String serialization of hash keys
    template.setHashKeySerializer(stringRedisSerializer);
     // Jackson serialization of values
    template.setValueSerializer(jackson2JsonRedisSerializer);
     // Jackson serialization of hash values
    template.setHashValueSerializer(jackson2JsonRedisSerializer);
    template.afterPropertiesSet();
    return template;
}
```

- Código de exemplo para usar o Spring Boot e o Lettuce para conectar-se ao Redis Cluster com pool de conexão

  a. Adicione a configuração do Redis ao arquivo de configuração **application.properties**.

  ```
  spring.redis.cluster.nodes=host:port
  spring.redis.cluster.max-redirects=3
  spring.redis.password=pwd
  spring.redis.lettuce.cluster.refresh.period=60
  spring.redis.lettuce.cluster.refresh.adaptive=true
  # Connection timeout.
  spring.redis.timeout=60s
  # Maximum number of connections in the connection pool. A negative value
  indicates no limit.
  spring.redis.lettuce.pool.max-active=50
  # Minimum number of idle connections in the connection pool.
  spring.redis.lettuce.pool.min-idle=5
  # Maximum number of idle connections in the connection pool.
  ```



```
spring.redis.lettuce.pool.max-idle=50
# Maximum time for waiting for connections in the connection pool. A
negative value indicates no limit.
spring.redis.lettuce.pool.max-wait=5000
# Interval for scheduling an eviction thread.
spring.redis.lettuce.pool.time-between-eviction-runs=2000
```

b. Classe de configuração RedisConfiguration do Redis (a atualização automatizada da topologia deve ser ativada).

```
@Bean
 public LettuceConnectionFactory lettuceConnectionFactory() {
     GenericObjectPoolConfig genericObjectPoolConfig = new
GenericObjectPoolConfig();
     genericObjectPoolConfig.setMaxIdle(maxIdle);
     genericObjectPoolConfig.setMinIdle(minIdle);
     genericObjectPoolConfig.setMaxTotal(maxActive);
     genericObjectPoolConfig.setMaxWait(Duration.ofMillis(maxWait));

genericObjectPoolConfig.setTimeBetweenEvictionRuns(Duration.ofMillis(time
BetweenEvictionRunsMillis));
     String[] nodes = clusterNodes.split(",");
     List<RedisNode> listNodes = new ArrayList();
     for (String node : nodes) {
         String[] ipAndPort = node.split(":");
         RedisNode redisNode = new RedisNode(ipAndPort[0],
Integer.parseInt(ipAndPort[1]));
         listNodes.add(redisNode);
     }
     RedisClusterConfiguration redisClusterConfiguration = new
RedisClusterConfiguration();
     redisClusterConfiguration.setClusterNodes(listNodes);
     redisClusterConfiguration.setPassword(password);
     redisClusterConfiguration.setMaxRedirects(maxRedirects);
      // Configure automated topology refresh.
     ClusterTopologyRefreshOptions topologyRefreshOptions =
ClusterTopologyRefreshOptions.builder()
         .enablePeriodicRefresh(Duration.ofSeconds(period)) // Refresh
the topology periodically.
         .enableAllAdaptiveRefreshTriggers() // Refresh the topology
based on events.
         .build();

     ClusterClientOptions clusterClientOptions =
ClusterClientOptions.builder()
         // Redis command execution timeout. Only when the command
execution times out will a reconnection be triggered using the new
topology.
         .timeoutOptions(TimeoutOptions.enabled(Duration.ofSeconds(period
)))
         .topologyRefreshOptions(topologyRefreshOptions)
         .build();
     LettuceClientConfiguration clientConfig =
LettucePoolingClientConfiguration.builder()
             .commandTimeout(Duration.ofSeconds(timeout))
             .poolConfig(genericObjectPoolConfig)
             .readFrom(ReadFrom.REPLICA_PREFERRED) // Preferentially
read data from the replicas.
             .clientOptions(clusterClientOptions)
             .build();
     LettuceConnectionFactory factory = new
LettuceConnectionFactory(redisClusterConfiguration, clientConfig);
     return factory;
 }

@Bean
public RedisTemplate<String, Object>
redisTemplate(LettuceConnectionFactory lettuceConnectionFactory) {
    lettuceConnectionFactory.setShareNativeConnection(false);
    RedisTemplate<String, Object> template = new RedisTemplate<>();
    template.setConnectionFactory(lettuceConnectionFactory);
```



```
     // Use Jackson2JsonRedisSerializer to replace the default
JdkSerializationRedisSerializer to serialize and deserialize the Redis
value.
    Jackson2JsonRedisSerializer<Object> jackson2JsonRedisSerializer =
new Jackson2JsonRedisSerializer<>(Object.class);
    ObjectMapper mapper = new ObjectMapper();
    mapper.setVisibility(PropertyAccessor.ALL,
JsonAutoDetect.Visibility.ANY);
    mapper.activateDefaultTyping(LaissezFaireSubTypeValidator.instance,
        ObjectMapper.DefaultTyping.NON_FINAL,
JsonTypeInfo.As.PROPERTY);
    jackson2JsonRedisSerializer.setObjectMapper(mapper);
    StringRedisSerializer stringRedisSerializer = new
StringRedisSerializer();
     // String serialization of keys
    template.setKeySerializer(stringRedisSerializer);
     // String serialization of hash keys
    template.setHashKeySerializer(stringRedisSerializer);
     // Jackson serialization of values
    template.setValueSerializer(jackson2JsonRedisSerializer);
     // Jackson serialization of hash values
    template.setHashValueSerializer(jackson2JsonRedisSerializer);
    template.afterPropertiesSet();
    return template;
}
```

host é o endereço IP/nome de domínio da instância do DCS, **port** é o número da porta da instância do DCS e **pwd** é a senha da instância do DCS. Especifique esses parâmetros conforme necessário antes de executar o código. O pool de conexão é recomendado. Ajuste parâmetros como **TimeOut**, **MaxTotal** (número máximo de conexões), **MinIdle** (número mínimo de conexões ociosas), **MaxIdle** (número máximo de conexões ociosas) e **MaxWait** (tempo máximo de espera) com base nos requisitos de serviço.

----Fim

4.3.4 Clientes em Python

Acesse uma instância do DCS Redis por meio do redis-py em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

NOTA

Use o redis-py para conectar-se a instâncias de cluster de nó único, principal/em espera e proxy e redis-py-cluster para conectar-se a instâncias de cluster do Redis.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executa o SO de Linux, certifique-se de que o ambiente de compilação Python tenha sido instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

A seguir, o CentOS é usado como um exemplo para descrever como acessar uma instância usando um cliente Python.

Passo 3 Acesse a instância do DCS Redis.

Se o sistema não fornecer Python, execute o seguinte comando **yum** para instalá-lo:

yum install python

NOTA

A versão do Python deve ser 3.6 ou posterior. Se a versão padrão do Python for anterior à 3.6, execute as seguintes operações para alterá-la:

1. Execute o comando **rm -rf python** para excluir o link simbólico do Python.
2. Execute o comando **ln -s python***X.X.X* **python** para criar outro link Python. No comando, *X.X.X* indica o número da versão do Python.

- Se a instância for uma instância de um único nó, principal/em espera ou Cluster de Proxy:
  a. Instale o Python e o redis-py.
     i. Se o sistema não fornecer Python, execute o seguinte comando **yum** para instalá-lo.
     ii. Execute o seguinte comando para baixar e descompactar o pacote redis-py:

     wget https://github.com/andymccurdy/redis-py/archive/master.zip

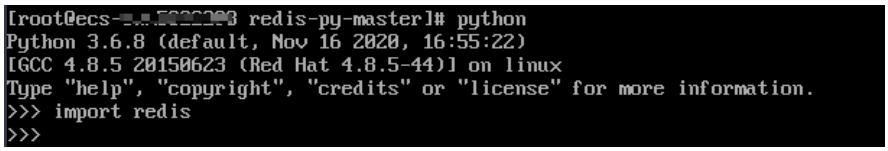
     unzip master.zip

     iii. Vá para o diretório onde o pacote redis-py descompactado está salvo e instale o redis-py.

     python setup.py install

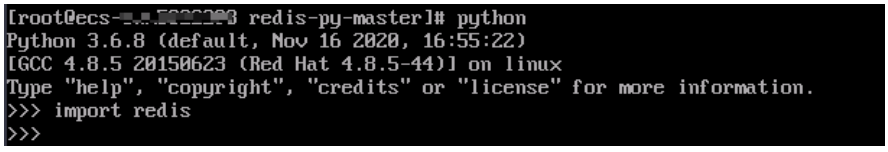
     Após a instalação, execute o comando **python**. O redis-py foi instalado com sucesso se a seguinte saída do comando for exibida:

     Figura 4-12 Executando o comando python

  b. Use o cliente redis-py para conectar-se à instância. Nas etapas a seguir, os comandos são executados no modo CLI. (Alternativamente, escreva os comandos em um script Python e, em seguida, execute o script.)
     i. Execute o comando **python** para entrar no modo CLI. Você entrou no modo CLI se a seguinte saída do comando for exibida:

     Figura 4-13 Entrando no modo CLI

ii. Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

```
r = redis.StrictRedis(host='XXX.XXX.XXX.XXX', port=6379,
password='******');
```

XXX.XXX.XXX.XXX indica o endereço IP/nome de domínio da instância do DCS e **6379** é um exemplo de número de porta da instância. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. ****** indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Você acessou a instância com êxito se a seguinte saída do comando for exibida. Digite comandos para executar operações de leitura e gravação no banco de dados.

Figura 4-14 Redis conectado com sucesso

- Se a instância for uma instância do Cluster do Redis:

  a. Instale o cliente redis-py-cluster.

  i. Baixe a versão lançada.

  wget https://github.com/Grokzen/redis-py-cluster/releases/download/2.1.3/redis-py-cluster-2.1.3.tar.gz

  ii. Descompacte o pacote.

  tar -xvf redis-py-cluster-2.1.3.tar.gz

  iii. Vá para o diretório onde o pacote redis-py-cluster descompactado está salvo e instale o redis-py-cluster.

  python setup.py install

  b. Acesse a instância do DCS Redis usando redis-py-cluster.

  Nas etapas a seguir, os comandos são executados no modo CLI. (Alternativamente, escreva os comandos em um script Python e, em seguida, execute o script.)

  i. Execute o comando **python** para entrar no modo CLI.

  ii. Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

```
>>> from rediscluster import RedisCluster

>>> startup_nodes = [{"host": "192.168.0.143", "port": "6379"}]

>>> rc = RedisCluster(startup_nodes=startup_nodes,
decode_responses=True)

>>> rc.set("foo", "bar")
True
>>> print(rc.get("foo"))
'bar'
```

----Fim

4.3.5 go-redis

Acesse uma instância do DCS Redis por meio do go-redis em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .

Procedimento

Passo 1 Exiba o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte**Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Um ECS do Windows é usado como exemplo.

Passo 3 Instale o Visual Studio Community 2017 no ECS.

Passo 4 Inicie o Visual Studio e crie um projeto. O nome do projeto pode ser personalizado. Neste exemplo, o nome do projeto é definido como **redisdemo**.

Passo 5 Importe o pacote de dependência do go-redis e digite **go get github.com/go-redis/redis** no terminal.

Passo 6 Escreva o seguinte código:

```
package main

import (
    "fmt"
    "github.com/go-redis/redis"
)

func main() {
    // Single-node
    rdb := redis.NewClient(&redis.Options{
        Addr:     "host:port",
        Password: "********", // no password set
        DB:       0,  // use default DB
    })

    val, err := rdb.Get("key").Result()
    if err != nil {
        if err == redis.Nil {
            fmt.Println("key does not exists")
            return
        }
        panic(err)
    }
    fmt.Println(val)

    //Cluster
    rdbCluster := redis.NewClusterClient(&redis.ClusterOptions{
        Addrs:    []string{"host:port"},
        Password: "********",
    })
```



```
    val1, err1 := rdbCluster.Get("key").Result()
    if err1 != nil {
        if err == redis.Nil {
            fmt.Println("key does not exists")
            return
        }
        panic(err)
    }
    fmt.Println(val1)
}
```

host:port são o endereço IP/nome do domínio e o número da porta da instância do DCS Redis. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. ******** indica a senha usada para efetuar login na instância do DCS Redis. Essa senha é definida durante a criação da instância do DCS Redis.

Passo 7 Execute o comando **go build -o test main.go** para empacotar o código em um arquivo executável, por exemplo, **test**.

⚠ CUIDADO

Para executar o pacote no SO de Linux, defina os seguintes parâmetros antes de empacotar:

set GOARCH=amd64

set GOOS=linux

Passo 8 Execute o comando **./test** para acessar a instância do DCS.

----Fim

4.3.6 hiredis em C++

Acesse uma instância do DCS Redis por meio de contratos em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

📖 NOTA

As operações descritas nesta seção se aplicam somente a instâncias de cluster de proxy, de nó único, mestre/em espera e de cluster de proxy. Para usar o C++ para se conectar a uma instância do Cluster do Redis, consulte **a descrição do cliente do C++ Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, verifique se o ambiente de compilação GCC foi instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

A seguir, o CentOS é usado como um exemplo para descrever como acessar uma instância em C++.

Passo 3 Instale o GCC, o Make e o Contractis.

Se o sistema não fornecer um ambiente de compilação, execute o seguinte comando **yum** para instalar o ambiente:

yum install gcc make

Passo 4 Execute o seguinte comando para fazer o download e descompactar o pacote hireis:

wget https://github.com/redis/hiredis/archive/master.zip

unzip master.zip

Passo 5 Vá para o diretório em que o pacote descompactado hireis é salvo, e compilar e instalar contratouis.

make

make install

Passo 6 Acesse a instância do DCS usando hiredis.

O seguinte descreve a autenticação de conexão e senha de contratouis. Para obter mais informações sobre como usar o hiredis, visite o site oficial do Redis.

1. Edite o código de exemplo para conexão a uma instância de DCS e salve o código e saia.

   vim connRedis.c

   Exemplo:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <hiredis.h>
int main(int argc, char **argv) {
     unsigned int j;
     redisContext *conn;
     redisReply *reply;
     if (argc < 3) {
             printf("Usage: example {instance_ip_address} 6379 {password}\n");
             exit(0);
     }
     const char *hostname = argv[1];
     const int port = atoi(argv[2]);
     const char *password = argv[3];
     struct timeval timeout = { 1, 500000 }; // 1.5 seconds
     conn = redisConnectWithTimeout(hostname, port, timeout);
     if (conn == NULL || conn->err) {
        if (conn) {
             printf("Connection error: %s\n", conn->errstr);
             redisFree(conn);
        } else {
             printf("Connection error: can't allocate redis context\n");
        }
     exit(1);
     }
     /* AUTH */
     reply = redisCommand(conn, "AUTH %s", password);
     printf("AUTH: %s\n", reply->str);
     freeReplyObject(reply);
```



```
    /* Set */
    reply = redisCommand(conn,"SET %s %s", "welcome", "Hello, DCS for
Redis!");
    printf("SET: %s\n", reply->str);
    freeReplyObject(reply);

    /* Get */
    reply = redisCommand(conn,"GET welcome");
    printf("GET welcome: %s\n", reply->str);
    freeReplyObject(reply);

    /* Disconnects and frees the context */
    redisFree(conn);
    return 0;
}
```

2. Execute o seguinte comando para compilar o código:

   gcc connRedis.c -o connRedis -I /usr/local/include/hiredis -lhiredis

   Se um erro for relatado, localize o diretório onde o arquivo **hiredis.h** é salvo e modifique o comando de compilação.

   Após a compilação, um arquivo executável **connRedis** é obtido.

3. Execute o seguinte comando para acessar a instância do DCS Redis escolhida:

   ./connRedis *{redis_ip_address}* 6379 *{password}*

   {redis_instance_address} indica o endereço IP/nome de domínio da instância DCS e **6379** é um número de porta de exemplo da instância DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. *{password}* indica a senha usada para efetuar login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

   Você acessou a instância com êxito se a seguinte saída do comando for exibida:

```
AUTH: OK
SET: OK
GET welcome: Hello, DCS for Redis!
```

AVISO

Se um erro for relatado, indicando que os arquivos de biblioteca do hireis não podem ser encontrados, execute os seguintes comandos para copiar arquivos relacionados para os diretórios do sistema e adicionar links dinâmicos:

mkdir /usr/lib/hiredis

cp /usr/local/lib/libhiredis.so.0.13 /usr/lib/hiredis/

mkdir /usr/include/hiredis

cp /usr/local/include/hiredis/hiredis.h /usr/include/hiredis/

echo '/usr/local/lib' >>;>>;/etc/ld.so.conf

ldconfig

Substitua os locais dos arquivos **so** e **.h** com os reais antes de executar os comandos.

----Fim

4.3.7 C#

Acesse uma instância do DCS Redis por meio do StackExchange do cliente C# em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, certifique-se de que o ambiente de compilação GCC tenha sido instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Um ECS do Windows é usado como exemplo.

Passo 3 Instale o Visual Studio Community 2017 no ECS.

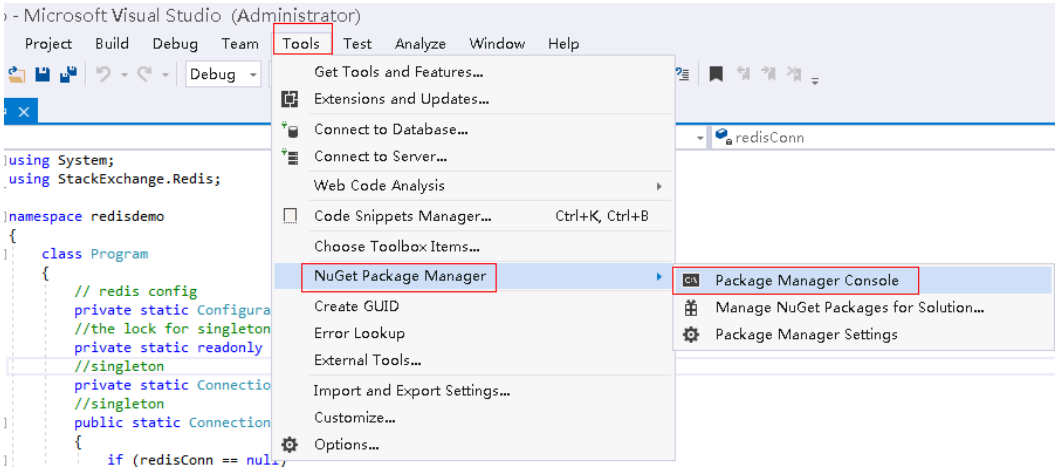
Passo 4 Inicie o Visual Studio 2017 e crie um projeto.

Defina o nome do projeto para **redisdemo**.

Passo 5 Instale o StackExchange Redis usando o gerenciador de pacotes do NuGet do Visual Studio.

Acesse o console do gerenciador de pacotes do NuGet de acordo com **Figura 4-15**, e insira **Install-Package StackExchange.Redis -*Version 2.2.79*** . (O número da versão é opcional).

Figura 4-15 Acessando o console do gerenciador de pacotes do NuGet



Passo 6 Escreva o código a seguir e use os métodos String Set e Get para testar a conexão.

```
using System;
using StackExchange.Redis;
```



```
namespace redisdemo
{
    class Program
    {
        // redis config
        private static ConfigurationOptions connDCS = ConfigurationOptions.Parse("
                            10.10
                            .38.233:6379
                        ,password=
                            ********
                        ,connectTimeout=2000");
        //the lock for singleton
        private static readonly object Locker = new object();
        //singleton
        private static ConnectionMultiplexer redisConn;
        //singleton
        public static ConnectionMultiplexer getRedisConn()
        {
            if (redisConn == null)
            {
                lock (Locker)
                {
                    if (redisConn == null || !redisConn.IsConnected)
                    {
                        redisConn = ConnectionMultiplexer.Connect(connDCS);
                    }
                }
            }
            return redisConn;
        }
        static void Main(string[] args)
        {
            redisConn = getRedisConn();
            var db = redisConn.GetDatabase();
            //set get
            string strKey = "Hello";
            string strValue = "DCS for Redis!";
            Console.WriteLine( strKey + ", " + db.StringGet(strKey));

            Console.ReadLine();
        }
    }
}
```

10.10.38.233:6379 contém um exemplo de endereço IP/nome de domínio e número de porta da instância do DCS Redis. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. ******** indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Passo 7 Execute o código. Você acessou a instância com êxito se a seguinte saída do comando for exibida:

```
Hello, DCS for Redis!
```

Para obter mais informações sobre outros comandos do StackExchange Redis, visite **StackExchange.Redis**.

----Fim

4.3.8 PHP

4.3.8.1 phpredis

Acesse uma instância do DCS Redis por meio do phpredis em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

NOTA

As operações descritas nesta seção se aplicam somente a instâncias de cluster de proxy, de nó único, principal/em espera e de cluster de proxy. Para usar o phpredis para se conectar a uma instância do Cluster do Redis, consulte **a descrição do phpredis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, verifique se o ambiente de compilação GCC foi instalado no ECS.

Procedimento

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

A seguir, o CentOS é usado como um exemplo para descrever como acessar uma instância através do phpredis.

Passo 3 Instale os componentes de compilação GCC-C++ e Make.

yum install gcc-c++ make

Passo 4 Instale o pacote de desenvolvimento PHP e a ferramenta CLI.

Execute o seguinte comando **yum** para instalar o pacote de desenvolvimento PHP:

yum install php-devel php-common php-cli

Após a conclusão da instalação, execute o seguinte comando para consultar a versão do PHP e verificar se a instalação foi bem-sucedida:

php --version

Passo 5 Instale o cliente phpredis.

1. Baixe o pacote fonte do phpredis.

   wget http://pecl.php.net/get/redis-5.3.7.tgz

   Esta versão é usada como exemplo. Para baixar clientes phpredis de outras versões, visite o site oficial do Redis ou PHP.

2. Descompacte o pacote fonte do phpredis.

   tar -zxvf redis-5.3.7.tgz

   cd redis-5.3.7

3. Comando antes da compilação.

   phpize

4. Configure o arquivo **php-config**.

   ./configure --with-php-config=/usr/bin/php-config

   A localização do arquivo varia dependendo do SO e do modo de instalação do PHP. É aconselhável localizar o diretório onde o arquivo é salvo antes da configuração.

   find / -name php-config

5. Compile e instale o cliente phpredis.

   make && make install

6. Após a instalação, adicione a configuração da **extension** no arquivo **php.ini** para fazer referência ao módulo Redis.

   vim /etc/php.ini

   Adicione a seguinte configuração:

   ```
   extension = "/usr/lib64/php/modules/redis.so"
   ```

   NOTA

   O arquivo **redis.so** pode ser salvo em um diretório diferente do **php.ini**. Execute o seguinte comando para localizar o diretório:

   find / -name php.ini

7. Salve a configuração e saia. Em seguida, execute o seguinte comando para verificar se a extensão tem efeito:

   php -m |grep redis

   Se a saída do comando contiver **redis**, o ambiente do cliente phpredis foi configurado.

Passo 6 Acesse a instância do DCS usando phpredis.

1. Edite um arquivo **redis.php**.

   ```
   <?php
       $redis_host = "{redis_instance_address}";
       $redis_port = 6379;
       $user_pwd = "{password}";
       $redis = new Redis();
       if ($redis->connect($redis_host, $redis_port) == false) {
          die($redis->getLastError());
       }
       if ($redis->auth($user_pwd) == false) {
           die($redis->getLastError());
       }
       if ($redis->set("welcome", "Hello, DCS for Redis!") == false) {
           die($redis->getLastError());
       }
       $value = $redis->get("welcome");
       echo $value;
       $redis->close();
   ?>
   ```

   {redis_instance_address} indica o endereço IP/nome de domínio da instância DCS e 6379 é um número de porta de exemplo da instância DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. *{password}* indica a senha usada para efetuar login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis. Se o acesso sem senha estiver habilitado, proteja a instrução **if** para autenticação de senha.

2. Execute o comando **php redis.php** para acessar a instância do DCS.

----Fim

4.3.8.2 Predis

Acesse uma instância do DCS Redis por meio do Predis em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, verifique se o ambiente de compilação do PHP foi instalado no ECS.

Procedimento

Passo 1 Exiba o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Passo 3 Instale o pacote de desenvolvimento PHP e a ferramenta CLI. Execute o seguinte comando **yum**:

yum install php-devel php-common php-cli

Passo 4 Após a conclusão da instalação, verifique o número da versão para garantir que a instalação seja bem-sucedida.

php --version

Passo 5 Baixe o pacote Predis para o diretório **/usr/share/php**.

1. Execute o seguinte comando para baixar o arquivo de origem do Predis:

   wget https://github.com/predis/predis/archive/refs/tags/v1.1.10.tar.gz

   NOTA

   Esta versão é usada como exemplo. Para baixar clientes Predis de outras versões, visite o site oficial do Redis ou PHP.

2. Execute os seguintes comandos para descompactar o pacote Predis de origem:

   tar -zxvf predis-1.1.10.tar.gz

3. Renomeie o diretório Predis descompactado **predis** e mova-o para **/usr/share/php/**.

   mv predis-1.1.10 predis

Passo 6 Edite um arquivo usado para conectar-se ao Redis.

- Exemplo de uso do **redis.php** para conectar-se a uma instância de Cluster DCS Redis de nó único, mestre/standby ou proxy:

```
<?php
    require 'predis/autoload.php';
```



```
    Predis\Autoloader::register();
    $client = new Predis\Client([
      'scheme' => 'tcp' ,
      'host'     => '{redis_instance_address}' ,
      'port'     => {port} ,
      'password' => '{password}'
    ]);
    $client->set('foo', 'bar');
    $value = $client->get('foo');
    echo $value;
?>
```

- Exemplo de código para usar **redis-cluster.php** para conectar-se ao Redis Cluster:

```
<?php
   require 'predis/autoload.php';
        $servers = array(
         'tcp://{redis_instance_address}:{port}'
        );
       $options = array('cluster' => 'redis');
       $client = new Predis\Client($servers, $options);
       $client->set('foo', 'bar');
       $value = $client->get('foo');
       echo $value;
?>
```

{redis_instance_address} indica o endereço IP real ou o nome de domínio da instância DCS e *{port}* é o número da porta real da instância DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. *{password}* indica a senha usada para efetuar login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis. Se for necessário acesso sem senha, exclua a linha que contém "senha".

Passo 7 Execute o comando **php redis.php** para acessar a instância do DCS.

----Fim

4.3.9 Node.js

Acesse uma instância do DCS Redis por meio do Node.js em um ECS na mesma VPC. Para obter mais informações sobre como usar outros clientes do Redis, visite **o site oficial do Redis**.

NOTA

As operações descritas nesta seção se aplicam somente a instâncias de cluster de proxy, principal/em espera e de nó único. Para usar o Node.js para se conectar a uma instância do Cluster do Redis, consulte **a descrição do cliente do Node.js Redis**.

Pré-requisitos

- Uma instância do DCS Redis foi criada e está no estado **Running**.
- Foi criado um ECS. Para obter detalhes sobre como criar um ECS, consulte **Comprando um ECS** .
- Se o ECS executar o SO de Linux, certifique-se de que o ambiente de compilação GCC tenha sido instalado no ECS.

Procedimento

- **Para servidores clientes que executam o Ubuntu (série Debian):**

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Passo 3 Instale o Node.js.

apt install nodejs-legacy

Se o comando anterior não funcionar, execute os seguintes comandos:

wget https://nodejs.org/dist/v0.12.4/node-v0.12.4.tar.gz --no-check-certificate

tar -xvf node-v4.28.5.tar.gz

cd node-v4.28.5

./configure

make

make install

NOTA

Após a conclusão da instalação, execute o comando **node --version** para consultar a versão do Node.js para verificar se a instalação foi bem-sucedida.

Passo 4 Instale o gerenciador de pacotes de nó (npm).

apt install npm

Passo 5 Instale o ioredis do cliente Redis.

npm install ioredis

Passo 6 Edite o script de exemplo para conexão com uma instância do DCS.

Adicione o seguinte conteúdo ao script **ioredisdemo.js**, incluindo informações sobre conexão e leitura de dados.

```
var Redis = require('ioredis');
var redis = new Redis({
  port: 6379,          // Redis port
  host: '192.168.0.196',   // Redis host
  family: 4,           // 4 (IPv4) or 6 (IPv6)
  password: '******',
  db: 0
});
redis.set('foo', 'bar');
redis.get('foo', function (err, result) {
  console.log(result);
});
// Or using a promise if the last argument isn't a function
redis.get('foo').then(function (result) {
  console.log(result);
});
// Arguments to commands are flattened, so the following are the same:
redis.sadd('set', 1, 3, 5, 7);
redis.sadd('set', [1, 3, 5, 7]);
// All arguments are passed directly to the redis server:
redis.set('key', 100, 'EX', 10);
```

host indica o exemplo de endereço IP/nome de domínio da instância do DCS e a *port* indica o número da porta da instância do DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta

conforme necessário. ****** indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Passo 7 Execute o script de exemplo para acessar a instância do DCS escolhida.

node ioredisdemo.js

----Fim

- **Para servidores clientes que executam CentOS (série Red Hat):**

Passo 1 Visualize o endereço IP/nome do domínio e o número da porta da instância do DCS Redis a ser acessada.

Para obter detalhes, consulte **Exibindo Detalhes da Instância** .

Passo 2 Acesse o ECS.

Passo 3 Instale o Node.js.

yum install nodejs

Se o comando anterior não funcionar, execute os seguintes comandos:

wget https://nodejs.org/dist/v0.12.4/node-v0.12.4.tar.gz --no-check-certificate

tar -xvf node-v0.12.4.tar.gz

cd node-v0.12.4

./configure

make

make install

NOTA

Após a conclusão da instalação, execute o comando **node --version** para consultar a versão do Node.js para verificar se a instalação foi bem-sucedida.

Passo 4 Instale o npm.

yum install npm

Passo 5 Instale o ioredis do cliente Redis.

npm install ioredis

Passo 6 Edite o script de exemplo para conexão com uma instância do DCS.

Adicione o seguinte conteúdo ao script **ioredisdemo.js**, incluindo informações sobre conexão e leitura de dados.

```
var Redis = require('ioredis');
var redis = new Redis({
  port: 6379,          // Redis port
  host: '192.168.0.196',   // Redis host
  family: 4,           // 4 (IPv4) or 6 (IPv6)
  password: '******',
  db: 0
});
redis.set('foo', 'bar');
redis.get('foo', function (err, result) {
```



```
  console.log(result);
});
// Or using a promise if the last argument isn't a function
redis.get('foo').then(function (result) {
  console.log(result);
});
// Arguments to commands are flattened, so the following are the same:
redis.sadd('set', 1, 3, 5, 7);
redis.sadd('set', [1, 3, 5, 7]);
// All arguments are passed directly to the redis server:
redis.set('key', 100, 'EX', 10);
```

host indica o exemplo de endereço IP/nome de domínio da instância do DCS e a *port* indica o número da porta da instância do DCS. Para obter detalhes sobre como obter o endereço IP/nome do domínio e a porta, consulte **Passo 1**. Altere o endereço IP/nome do domínio e a porta conforme necessário. ****** indica a senha usada para fazer login na instância do DCS Redis escolhida. Essa senha é definida durante a criação da instância do DCS Redis.

Passo 7 Execute o script de exemplo para acessar a instância do DCS escolhida.

node ioredisdemo.js

----Fim

4.4 Acesso da CLI da Web a uma instância do DCS Redis 4.0/5.0

Acesse um DCS Redis instância por meio da Web CLI. Esta função é suportada apenas por instâncias do DCS Redis 4.0/5.0 e não por instâncias do DCS Redis 3.0.

NOTA

- Não insira informações confidenciais na CLI da Web para evitar divulgação.
- Se o valor estiver vazio, é devolvido **nil** após a execução do comando **GET**.

Pré-requisitos

A instância do DCS Redis 4.0/5.0 que você deseja acessar por meio da Web CLI está no estado **Running**.

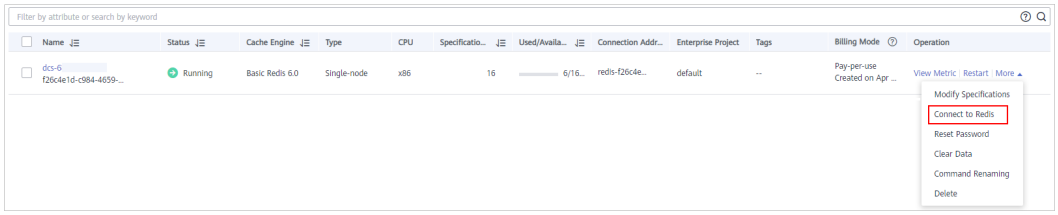
Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**. Na coluna **Operation** da instância, escolha **More** > **Connect to Redis**, conforme mostrado na **Figura 4-16**.

Figura 4-16 Acessando a CLI da Web



Passo 4 Digite a senha de acesso da instância do DCS. Na CLI da Web, selecione o banco de dados atual do Redis, insira um comando do Redis na caixa de comando e pressione **Enter**.

NOTA

Se nenhuma operação for realizada por mais de 5 minutos, a conexão expira. Você deve inserir a senha de acesso para se conectar à instância novamente.

----Fim

5 Acessando uma Instância do Memcached de DCS

5.1 telnet

Acesse uma instância do Memcached DCS usando telnet em um ECS na mesma VPC.

Pré-requisitos

- A instância do Memcached DCS que você deseja acessar está no estado **Running**.
- Foi criado um ECS no qual o cliente foi instalado. Para obter detalhes sobre como criar os ECS, consulte o *Guia do usuário do Elastic Cloud Server*.

  NOTA

  Um ECS pode se comunicar com uma instância de DCS que pertence à mesma VPC e está configurada com o mesmo grupo de segurança.

  - Se a instância do ECS e do DCS estiverem em VPCs diferentes, estabeleça uma conexão de peering de VPC para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **O DCS oferece suporte ao acesso entre VPC?**
  - Se grupos de segurança diferentes tiverem sido configurados para a instância do ECS e do DCS, defina regras de grupo de segurança para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **Como configurar um grupo de segurança?**
- Todas as anotações no código de exemplo foram excluídas.
- Todas as linhas de comando e blocos de código são codificados em UTF-8. Usar outro esquema de codificação causará problemas de compilação ou mesmo falhas de comando.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, clique no nome da instância do Memcached DCS que deseja acessar. Obtenha o endereço IP e o número da porta da instância.

Passo 5 Acesse a instância do Memcached DCS escolhida.

1. Acesse o ECS.
2. Execute o seguinte comando para verificar se o telnet está instalado no ECS:

   which telnet

   Se o diretório no qual o telnet está instalado for exibido, o telnet foi instalado no ECS. Se o diretório de instalação do cliente não for exibido, instale o telnet manualmente.

   NOTA

   - Se o telnet não tiver sido instalado no Linux, execute o comando **yum -y install telnet** para o instalar.
   - No SO de Windows, escolha **Start** > **Control Panel** > **Programs** > **Programs and Features** > **Turn Windows features on or off**, e habilitar o telnet.

3. Execute o seguinte comando para acessar a instância do Memcached DCS escolhida:

   telnet ***{ip or domain name} {port}***

   Neste comando: ***{ip address or domain name}*** indica o endereço IP ou nome de domínio da instância do Memcached DCS. ***{port}*** indica o número da porta da instância do Memcached DCS. Tanto o endereço IP ou nome de domínio e o número da porta são obtidos em **Passo 4**.

   Quando você tiver acessado com êxito a instância do Memcached do DCS escolhida, informações semelhantes às seguintes serão exibidas:

   ```
   Trying XXX.XXX.XXX.XXX...
   Connected to XXX.XXX.XXX.XXX.
   Escape character is '^]'.
   ```

   NOTA

   - Se **Password Protected** não estiver habilitado para a instância, execute os seguintes comandos diretamente depois que a instância for acessada com êxito.
   - Se **Password Protected** estiver ativado para a instância, as tentativas de executar operações na instância resultarão na mensagem "ERROR autenticação necessária", indicando que você não tem as permissões necessárias. Nesse caso, digite **auth** ***username@password*** para autenticar primeiro. *username* e *password* são os usados para acessar a instância do Memcached DCS.

   Exemplo de comandos para usar a instância do Memcached DCS (linhas em negrito são os comandos e as outras linhas são a saída do comando):

   ```
                                   set hello 0 0 6
                                   world!
   STORED
                                   get hello
   VALUE hello 0 6
   world!
   END
   ```

----Fim

5.2 Java

Acesse uma instância do DCS Memcached usando um cliente Java em um ECS na mesma VPC.

Pré-requisitos

- A instância do Memcached DCS que você deseja acessar está no estado **Running**.
- Foi criado um ECS no qual o cliente foi instalado. Para obter detalhes sobre como criar os ECS, consulte o *Guia do usuário do Elastic Cloud Server*.

  NOTA

  Um ECS pode se comunicar com uma instância de DCS que pertence à mesma VPC e está configurada com o mesmo grupo de segurança.

  - Se a instância do ECS e do DCS estiverem nas VPC diferentes, estabeleça uma conexão de peering de VPC para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **O DCS oferece suporte ao acesso entre VPC?**
  - Se grupos de segurança diferentes tiverem sido configurados para a instância do ECS e do DCS, defina regras de grupo de segurança para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **Como configurar um grupo de segurança?**
- O kit de desenvolvimento Java (JDK) e ambientes de desenvolvimento integrados comuns (os IDE), como o Eclipse, foram instalados no ECS.
- Você obteve o pacote de dependências **spymemcached-*x.y.z*.jar**.

  NOTA

  x.y.z indica a versão do pacote de dependência. Recomenda-se a versão mais recente.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, clique no nome da instância do Memcached DCS que deseja acessar. Obtenha o endereço IP ou o nome de domínio e o número da porta da instância.

Passo 5 Carregue o pacote de dependências **spymemcached-*x.y.z*.jar** obtido no ECS criado.

Passo 6 Acesse o ECS.

Passo 7 Crie um projeto Java no Eclipse e importe o pacote de dependência **spymemcached-*x.y.z*.jar**. O nome do projeto é personalizável.

Passo 8 Crie uma classe **ConnectMemcached1**, copie o seguinte código Java para a classe e modifique o código.

- Exemplo de código para o modo de senha

  Alterar *IP ou nome de domínio:porta* para o endereço IP e o número da porta obtidos em **Passo 4**. Definir *Nome de usuário* e *Senha* respectivamente para o nome de usuário e a senha da instância do Memcached.

```
//Connect to the encrypted Memcached code using Java.
import java.io.IOException;
import java.util.concurrent.ExecutionException;

import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
```



```
import net.spy.memcached.ConnectionFactoryBuilder.Protocol;
import net.spy.memcached.MemcachedClient;
import net.spy.memcached.auth.AuthDescriptor;
import net.spy.memcached.auth.PlainCallbackHandler;
import net.spy.memcached.internal.OperationFuture;

public class ConnectMemcached1
{
    public static void main(String[] args)
    {
        final String connectionaddress = "
                                    ip or domain name:port
                                ";
        final String username = "
                                    userName
                                ";//Indicates the username.
        final String password = "
                                    password
                                ";//Indicates the password.
        MemcachedClient client = null;
        try
        {
            AuthDescriptor authDescriptor =
                new AuthDescriptor(new String[] {"PLAIN"}, new
PlainCallbackHandler(username,
                        password));
            client = new MemcachedClient(
                    new
ConnectionFactoryBuilder().setProtocol(Protocol.BINARY)
                            .setAuthDescriptor(authDescriptor)
                            .build(),
                    AddrUtil.getAddresses(connectionaddress));
            String key = "memcached";//Stores data with the key being
memcached in Memcached.
            String value = "Hello World";//The value is Hello World.
            int expireTime = 5; //Specifies the expiration time, measured in
seconds. The countdown starts from the moment data is written. After the
expireTime elapses, the data expires and can no longer be read.
            doExcute(client, key, value, expireTime);//Executes the operation.
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }

    /**
     *Method of writing data to Memcached
     */
    private static void doExcute(MemcachedClient client, String key, String
value, int expireTime)
    {
        try
        {
            OperationFuture<Boolean> future = client.set(key, expireTime,
value);
            future.get();//spymemcached set () is asynchronous. future.get ()
waits until the cache.set () operation is completed, or does not need to
wait. You can select based on actual requirements.
            System.out.println("The Set operation succeeded.");
            System.out.println("Get operation:" + client.get(key));
            Thread.sleep(6000);//Waits for 6000 ms, that is, 6s. Then the
data expires and can no longer be read.
            System.out.println("Perform the Get operation 6s later:" +
client.get(key));

        }
        catch (InterruptedException e)
        {
```



```
            e.printStackTrace();
        }
        catch (ExecutionException e)
        {
            e.printStackTrace();
        }
        if (client != null)
        {
            client.shutdown();
        }
    }
}
```

- Código de exemplo para o modo livre de senha

  Altere **ip address or domain name:port** para o endereço IP e o número da porta obtidos em **Passo 4**.

```
//Connect to the password-free Memcached code using Java.
import java.io.IOException;
import java.util.concurrent.ExecutionException;

import net.spy.memcached.AddrUtil;
import net.spy.memcached.BinaryConnectionFactory;
import net.spy.memcached.MemcachedClient;
import net.spy.memcached.internal.OperationFuture;

public class ConnectMemcached
{
    public static void main(String[] args)
    {
        final String connectionaddress = "ip or domain name:port";
        MemcachedClient client = null;
        try
        {
            client = new MemcachedClient(new BinaryConnectionFactory(),
AddrUtil.getAddresses(connectionaddress));
            String key = "memcached";//Stores data with the key being
memcached in Memcached.
            String value = "Hello World";//The value is Hello World.
            int expireTime = 5; //Specifies the expiration time, measured in
seconds. The countdown starts from the moment data is written. After the
expireTime elapses, the data expires and can no longer be read.
            doExcute(client, key, value, expireTime);//Executes the operation.
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }

    /**
     *Method of writing data to Memcached
     */
    private static void doExcute(MemcachedClient client, String key, String
value, int expireTime)
    {
        try
        {
            OperationFuture<Boolean> future = client.set(key, expireTime,
value);
            future.get();//spymemcached set () is asynchronous. future.get ()
waits until the cache.set () operation is completed, or does not need to
wait. You can select based on actual requirements.
            System.out.println("The Set operation succeeded.");
            System.out.println("Get operation:" + client.get(key));
            Thread.sleep(6000);//Waits for 6000 ms, that is, 6s. Then the
data expires and can no longer be read.
            System.out.println("Perform the Get operation 6s later:" +
client.get(key));
```



```
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        catch (ExecutionException e)
        {
            e.printStackTrace();
        }
        if (client != null)
        {
            client.shutdown();
        }
    }
}
```

Passo 9 Execute o método **main**. O seguinte resultado é exibido na janela **Console** do Eclipse:

```
The Set operation succeeded.
Get operation: Hello World
Perform the Get operation 6s later: null
```

----Fim

5.3 Python

Acesse uma instância do DCS Memcached usando Python em um ECS na mesma VPC.

Pré-requisitos

- A instância do Memcached DCS que você deseja acessar está no estado **Running**.
- Acesse o ECS. Para obter detalhes sobre como criar os ECS, consulte o *Guia do usuário do Elastic Cloud Server*.

  NOTA

  Um ECS pode se comunicar com uma instância de DCS que pertence à mesma VPC e está configurada com o mesmo grupo de segurança.

  - Se a instância de ECS e DCS estiver nas VPC diferentes , estabeleça uma conexão de peering de VPC para alcançar a conectividade de rede entre a instância de ECS e DCS. Para obter detalhes, consulte **O DCS oferece suporte ao acesso entre VPC?**
  - Se grupos de segurança diferentes tiverem sido configurados para a instância do ECS e do DCS, defina regras de grupo de segurança para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **Como configurar um grupo de segurança?**
- O Python foi instalado no ECS. A versão recomendada é a 2.7.6 ou posterior.
- Você obteve o pacote de dependência **python-binary-memcached-x.y.z.zip**.

  NOTA

  x.y.z indica a versão do pacote de dependência. Recomenda-se a versão mais recente.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, clique no nome da instância do Memcached DCS que deseja acessar. Obtenha o endereço IP ou o nome de domínio e o número da porta da instância.

Passo 5 Carregue o pacote de dependência obtido (por exemplo, o pacote **python-binary-memcached-*x.y.z*.zip**) no ECS criado.

Passo 6 Acesse o ECS.

Passo 7 Execute os seguintes comandos para instalar o pacote de dependência:

unzip -xzvf python-binary-memcached-x.y.z.zip

cd python-binary-memcached-x.y.z

python setup.py install

NOTA

Se um erro for relatado durante a instalação, use o método de instalação **apt** ou **yum**. Por exemplo, para instalar o pacote de dependência usando o método **apt**, execute os seguintes comandos:

apt install python-pip;

pip install python-binary-memcached;

Passo 8 Crie um arquivo Python chamado **dcs_test.py**, copie o seguinte código Python para o arquivo e modifique o código.

- Exemplo de código para o modo de senha

  Alterar *ip ou nome de domínio:porta* ao endereço IP ou nome de domínio e número de porta obtidos em **Passo 4**. Definir *Nome de usuário* e *Senha* respectivamente para o nome de usuário e a senha da instância do Memcached.

```
import bmemcached
client = bmemcached.Client(('
                                        ip or domain name:port
                                    '), '
                                        userName
                                    ', '
                                        password
                                    ')
print "set('key', 'hello world!')"
print client.set('key', 'hello world!')
print "get('key')"
print client.get('key')
```

- Código de exemplo para o modo sem senha

  Altere **ip address or domain name:port** para o endereço IP e o número da porta obtidos em **Passo 4**.

```
import bmemcached
client = bmemcached.Client('ip or domain name:port')
print "set('key', 'hello world!')"
print client.set('key', 'hello world!')
print "get('key')"
print client.get('key')
```

Passo 9 Execute o arquivo **dcs_test.py**. O resultado seguinte é exibido .

```
# python test.py
set('key', 'hello world!')
True
get('key')
hello world!
```

----Fim

5.4 C++

Acesse uma instância do DCS Memcached usando um cliente C++ em um ECS na mesma VPC.

Pré-requisitos

- A instância do Memcached DCS que você deseja acessar está no estado **Running**.
- Acesse o ECS. Para obter detalhes sobre como criar os ECS, consulte o *Guia do usuário do Elastic Cloud Server*.

  NOTA

  Um ECS pode se comunicar com uma instância de DCS que pertence à mesma VPC e está configurada com o mesmo grupo de segurança.

  - Se a instância do ECS e do DCS estiverem nas VPC diferentes, estabeleça uma conexão de peering de VPC para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **O DCS oferece suporte ao acesso entre VPC?**
  - Se grupos de segurança diferentes tiverem sido configurados para a instância do ECS e do DCS, defina regras de grupo de segurança para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **Como configurar um grupo de segurança?**
- O GCC foi instalado no ECS. A versão recomendada é a 4.8.4 ou posterior.
- Você obteve o pacote de dependências **libmemcached-x.y.z.tar.gz**.

  NOTA

  x.y.z indica a versão do pacote de dependência. Recomenda-se a versão mais recente.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, clique no nome da instância do Memcached DCS que deseja acessar. Obtenha o endereço IP ou o nome de domínio e o número da porta da instância.

Passo 5 Carregue o pacote de dependência **libmemcached-*x.y.z*.tar.gz** obtido para o ECS criado.

Passo 6 Acesse o ECS.

Passo 7 Instalar os pacotes de dependência SASL relacionados.

Para sistemas operacionais da série Debian: **apt install libsasl2-dev cloog.ppl**

Para sistemas operacionais da série Red Hat: **yum install cyrus-sasl***

Passo 8 Execute os seguintes comandos para instalar o pacote de dependência:

tar -xzvf libmemcached-x.y.z.tar.gz

cd libmemcached-x.y.z

./configure --enable-sasl

make

make install

Passo 9 Crie um arquivo chamado **build.sh** e copie o seguinte código para o arquivo.

```
g++ -o dcs_sample dcs_sample.cpp -lmemcached -std=c++0x -lpthread -lsasl2
```

NOTA

Se o arquivo **libmemcached.so.11** não puder ser encontrado durante a compilação, execute o comando **find** para encontrar o arquivo e copie o arquivo para o diretório **/usr/lib**.

Passo 10 Crie um arquivo chamado **dcs_sample.cpp**, copie o seguinte código C++ para o arquivo e modifique o código.

- Exemplo de código para o modo de senha

  Alterar *ip ou nome de domínio* e *Porta* ao endereço IP ou nome de domínio e número de porta obtidos em **Passo 4**. Definir *Nome de usuário* e *Senha* respectivamente para o nome de usuário e a senha da instância do Memcached.

```
#include <iostream>
#include <string>
#include <libmemcached/memcached.h>
using namespace std;

#define IP  "
                                        ip

                                            or domain name

                                    "
#define PORT  "
                                        port
                                    "
#define USERNAME "
                                        userName
                                    "
#define PASSWORD "
                                        password
                                    "
memcached_return result;

memcached_st * init()
{
    memcached_st *memcached = NULL;
    memcached_server_st *cache;
    memcached = memcached_create(NULL);
    cache = memcached_server_list_append(NULL, IP, PORT, &result);

    sasl_client_init(NULL);
         memcached_set_sasl_auth_data(memcached, USERNAME, PASSWORD);

memcached_behavior_set(memcached,MEMCACHED_BEHAVIOR_BINARY_PROTOCOL,1);
         memcached_server_push(memcached,cache);
         memcached_server_list_free(cache);
            return memcached;
}

int main(int argc, char *argv[])
{
         memcached_st *memcached=init();
    string key = "memcached";
    string value = "hello world!";
    size_t value_length = value.length();
```



```
    int expire_time = 0;
    uint32_t flag = 0;

    result =
memcached_set(memcached,key.c_str(),key.length(),value.c_str(),value.length(),
expire_time,flag);
    if (result != MEMCACHED_SUCCESS){
      cout <<"set data failed: " << result << endl;
      return -1;
    }
    cout << "set succeed, key: " << key << ", value: " << value << endl;
    cout << "get key:" << key << endl;
    char* result =
memcached_get(memcached,key.c_str(),key.length(),&value_length,&flag,&result);
    cout << "value:" << result << endl;

    memcached_free(memcached);
    return 0;
}
```

- Código de exemplo para o modo livre de senha

  Altere *ip or domain name* para o endereço IP ou nome de domínio e número de porta obtido em **Passo 4**.

```
#include <iostream>
#include <string>
#include <libmemcached/memcached.h>
using namespace std;

#define IP  "ip
                                  or domain name"
#define PORT  port
memcached_return result;

memcached_st * init()
{
    memcached_st *memcached = NULL;
    memcached_server_st *cache;
    memcached = memcached_create(NULL);
    cache = memcached_server_list_append(NULL, IP, PORT, &result);
         memcached_server_push(memcached,cache);
    memcached_server_list_free(cache);
         return memcached;
}

int main(int argc, char *argv[])
{
         memcached_st *memcached=init();
    string key = "memcached";
    string value = "hello world!";
    size_t value_length = value.length();
    int expire_time = 0;
    uint32_t flag = 0;

    result =
memcached_set(memcached,key.c_str(),key.length(),value.c_str(),value.length(),
expire_time,flag);
    if (result != MEMCACHED_SUCCESS){
      cout <<"set data failed: " << result << endl;
      return -1;
    }
    cout << "set succeed, key: " << key << " ,value: " << value << endl;
    cout << "get key:" << key << endl;
    char* result =
memcached_get(memcached,key.c_str(),key.length(),&value_length,&flag,&result);
    cout << "value:" << result << endl;

    memcached_free(memcached);
    return 0;
}
```


Passo 11 Execute os seguintes comandos para compilar o código-fonte :

chmod 700 build.sh

./build.sh

O arquivo binário **dcs_sample** é gerado.

Passo 12 Execute o seguinte comando para acessar a instância do Memcached DCS escolhida:

./dcs_amostra

```
set succeed, key: memcached ,value: hello world!
get key:memcached
value:hello world!
```

----Fim

5.5 PHP

Acesse uma instância do DCS Memcached no PHP em um ECS na mesma VPC.

Pré-requisitos

- A instância do Memcached DCS que você deseja acessar está no estado **Running**.
- Efetue login no ECS. Para obter detalhes sobre como criar os ECS, consulte o *Guia do usuário do Elastic Cloud Server*.

  NOTA

  Um ECS pode se comunicar com uma instância de DCS que pertence à mesma VPC e está configurada com o mesmo grupo de segurança.

  - Se a instância do ECS e do DCS estiverem nas VPC diferentes, estabeleça uma conexão de peering de VPC para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **O DCS oferece suporte ao acesso entre VPC?**
  - Se grupos de segurança diferentes tiverem sido configurados para a instância do ECS e do DCS, defina regras de grupo de segurança para obter conectividade de rede entre o ECS e a instância do DCS. Para obter detalhes, consulte **Como configurar um grupo de segurança?**

Os SO da série Red Hat

A seguir, o CentOS 7.0 é usado como um exemplo para descrever como instalar um cliente PHP e usá-lo para acessar uma instância do Memcached DCS. O procedimento também é aplicável a um cliente PHP rodando o SO de Red Hat ou Fedora.

Passo 1 Instale os componentes de compilação GCC-C++ e Make.

yum install gcc-c++ make

Passo 2 Instalar pacotes SASL relacionados.

yum install cyrus-sasl*

Passo 3 Instale a biblioteca libMemcached.

A instalação da biblioteca libMemcached requer parâmetros de autenticação SASL. Portanto, você não pode instalar a biblioteca executando o comando **yum**.

wget https://launchpad.net/libmemcached/1.0/1.0.18/+download/libmemcached-1.0.18.tar.gz

tar -xvf libmemcached-1.0.18.tar.gz

cd libmemcached-1.0.18

./configure --prefix=/usr/local/libmemcached --enable-sasl

make && make install

NOTA

Antes de instalar a biblioteca libMemcached, instale os componentes GCC-C++ e SASL. Caso contrário, um erro será relatado durante a compilação. Depois de resolver o erro, execute o comando **make clean** e, em seguida, execute o comando **make** novamente.

Passo 4 Instalar o ambiente PHP.

yum install php-devel php-common php-cli

AVISO

PHP 7.x não suporta autenticação SASL. Use o PHP 5.6. Se a versão do yum php não for 5.6, baixe uma da Internet.

Passo 5 Instale o cliente Memcached.

Observe que você deve adicionar um parâmetro usado para habilitar o SASL ao executar o comando **configure**.

wget http://pecl.php.net/get/memcached-2.1.0.tgz

tar zxvf memcached-2.1.0.tgz

cd memcached-2.1.0

phpize

./configure --with-libmemcached-dir=/usr/local/libmemcached --enable-memcached-sasl

make && make install

Passo 6 Modifique o arquivo **php.ini**.

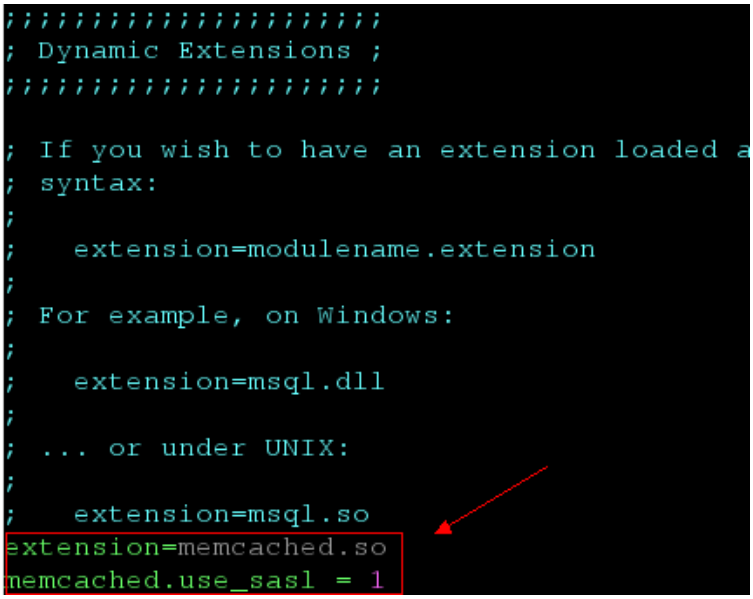
Execute o comando **find** ou **locate** para encontrar o arquivo **php.ini**.

find / -name php.ini

Adicione as duas linhas a seguir ao arquivo **php.ini**:

```
extension=memcached.so
memcached.use_sasl = 1
```


Figura 5-1 Modificando o arquivo **php.ini**



Passo 7 Acessar uma instância do Memcached DCS.

Crie um arquivo **memcached.php** e adicione o seguinte conteúdo ao arquivo:

```
<?php
    $connect = new Memcached; //Declares a Memcached connection.
    $connect->setOption(Memcached::OPT_COMPRESSION, false); //Disables
compression.
    $connect->setOption(Memcached::OPT_BINARY_PROTOCOL, true); //Uses the binary
protocol.
    $connect->setOption(Memcached::OPT_TCP_NODELAY, true); //Disables the TCP
network delay policy.
    $connect->addServer('{memcached_instance_ip}', 11211); //Specifies the
instance IP address and port number.
    $connect->setSaslAuthData('{username}', '{password}'); //If password-free
access is enabled for the instance, delete or comment out this line.
    $connect->set("DCS", "Come on!");
    echo 'DCS: ',$connect->get("DCS");
    echo "\n";
    $connect->quit();
?>
```

Salve e execute o arquivo **memcached.php**. O seguinte resultado é exibido.

```
[root@testphpmemcached ~]# php memcached.php
    DCS: Come on!
[root@testphpmemcached ~]#
```

----Fim

Sistemas operacionais da série Debian

O seguinte usa o SO de Ubuntu como um exemplo para descrever como instalar um cliente PHP e usá-lo para acessar uma instância do DCS Memcached.

Passo 1 Instale os componentes de compilação GCC e Make.

apt install gcc make

Passo 2 Instalar o ambiente PHP.

O PHP 5.x é recomendado para uma melhor compatibilidade com a autenticação SASL.

Execute os seguintes comandos para adicionar a fonte da imagem do PHP de uma versão anterior e, em seguida, instale os pacotes **php.5.6** e **php.5.6-dev**:

apt-get install -y language-pack-en-base;

LC_ALL=en_US.UTF-8;

add-apt-repository ppa:ondrej/php;

apt-get update;

apt-get install php5.6 php5.6-dev;

Depois que a instalação estiver completa, execute o comando **php -version** para verificar a versão do PHP. Se o seguinte resultado for exibido, a versão do PHP é 5.6, indicando que o PHP 5.6 foi instalado com sucesso.

```
root@dcs-nodelete:/etc/apt# php -version
PHP 5.6.36-1+ubuntu16.04.1+deb.sury.org+1 (cli)
Copyright (c) 1997-2016 The PHP Group
```

NOTA

Para desinstalar o PHP, execute os seguintes comandos:

apt install aptitude -y

aptitude purge `dpkg -l | grep php| awk '{print $2}' |tr "\n" " "`

Passo 3 Instale o componente SASL.

apt install libsasl2-dev cloog.ppl

Passo 4 Instale a biblioteca libMemcached.

wget https://launchpad.net/libmemcached/1.0/1.0.18/+download/libmemcached-1.0.18.tar.gz

tar -xvf libmemcached-1.0.18.tar.gz

cd libmemcached-1.0.18

./configure --prefix=/usr/local/libmemcached

make && make install

NOTA

Antes de instalar a biblioteca libMemcached, instale os componentes GCC-C++ e SASL. Caso contrário, um erro será relatado durante a compilação. Depois de resolver o erro, execute o comando **make clean** e, em seguida, execute o comando **make** novamente.

Passo 5 Instale o cliente Memcached.

Instale o componente zlib.

apt install zlib1g.dev

Observe que você deve adicionar um parâmetro usado para habilitar o SASL ao executar o comando **configure**.

wget http://pecl.php.net/get/memcached-2.2.0.tgz;

tar zxvf memcached-2.2.0.tgz;

cd memcached-2.2.0;

phpize5.6;

./configure --with-libmemcached-dir=/usr/local/libmemcached --enable-memcached-sasl;

make && make install;

Passo 6 Modifique o arquivo **pdo.ini**.

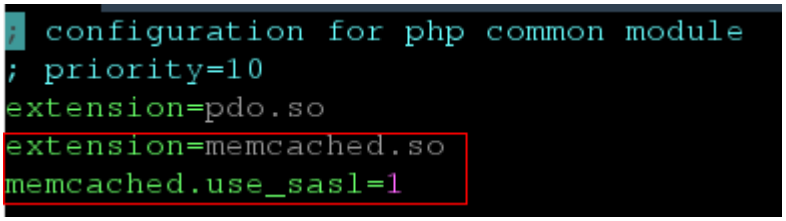
Execute o seguinte comando para localizar o arquivo **pdo.ini**:

find / -name pdo.ini

Por padrão, o arquivo **pdo.ini** é armazenado no diretório **/etc/php/5.6/mods-available**. Adicione as duas linhas a seguir ao arquivo **php.ini**:

```
extension=memcached.so
memcached.use_sasl = 1
```

Figura 5-2 Modificando o arquivo **pdo.ini**



Passo 7 Acesse uma instância do Memcached DCS.

Crie um arquivo **memcached.php** e adicione o seguinte conteúdo ao arquivo:

```
<?php
    $connect = new Memcached; //Declares a Memcached connection.
    $connect->setOption(Memcached::OPT_COMPRESSION, false); //Disables
compression.
    $connect->setOption(Memcached::OPT_BINARY_PROTOCOL, true); //Uses the binary
protocol.
    $connect->setOption(Memcached::OPT_TCP_NODELAY, true); //Disables the TCP
network delay policy.
    $connect->addServer('{memcached_instance_ip}', 11211); //Specifies the
instance IP address and port number.
    $connect->setSaslAuthData('{username}', '{password}'); //If password-free
access is enabled for the instance, delete or comment out this line.
    $connect->set("DCS", "Come on!");
    echo 'DCS: ',$connect->get("DCS");
    echo "\n";
    $connect->quit();
?>
```

Salve e execute o arquivo **memcached.php**. O seguinte resultado é exibido.

```
[root@dcs-nodelete ~]# php memcached.php
    DCS: Come on!
[root@dcs-nodelete ~]#
```

----Fim

6 Operando instâncias de DCS

6.1 Exibindo Detalhes da Instância

No console do DCS, você pode exibir os detalhes da instância do DCS.

NOTA

O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Pesquise instâncias de DCS usando qualquer um dos seguintes métodos:

- Busca por palavra-chave.

  Digite uma palavra-chave para pesquisar.

- Selecione atributos e insira suas palavras-chave para pesquisar.

  Atualmente, você pode pesquisar por nome, especificação, ID, endereço IP, AZ, status, tipo de instância, mecanismo de cache, projeto empresarial, modo de faturamento e tags.

  Por exemplo, para filtrar instâncias de DCS por mecanismo de cache ou versão do mecanismo de cache, clique na caixa de pesquisa, escolha **Cache Engine** e, em seguida, escolha **Redis**, **Redis 3.0**, **Redis 4.0**, **Redis 5.0**, ou **Memcached**.

Para obter mais informações sobre como pesquisar, clique no ponto de interrogação à direita da caixa de pesquisa.

Passo 5 Clique no nome da instância do DCS para exibir mais detalhes sobre a instância do DCS. **Tabela 6-1** descreve os parâmetros.

Tabela 6-1 Parâmetros na página Informações Básicas de uma instância de DCS

Seção	Parâmetro	Descrição
Detalhes da Instância	Nome	Nome da instância escolhida. Para modificar o nome da ocorrência, clique no ícone de .
	Status	Estado da instância escolhida.
	ID	ID da instância escolhida.
	Mecanismo de cache	Mecanismo de cache usado pela instância do DCS, que pode ser Redis ou Memcached. Se o mecanismo de cache for o Redis, ele será seguido pelo número da versão, por exemplo, Redis 3.0.
	Tipos de instância	Tipo da instância selecionada. Atualmente, os tipos suportados incluem nó único, principal/em espera, cluster de proxy, e cluster do Redis.
	Tamanho do cache	Especificação da instância escolhida.
	Memória Usada/ Disponível (MB)	O espaço de memória usado e o espaço máximo de memória disponível da instância escolhida. O espaço de memória utilizado inclui: ● Tamanho dos dados armazenados na instância do DCS ● Tamanho dos buffers do Redis-server (incluindo buffer do cliente e repl-backlog) e estruturas de dados internas
	CPU	Arquitetura da CPU da instância escolhida. Esse parâmetro é exibido somente para instâncias do DCS Redis.
	Projeto corporativo	Projeto empresarial ao qual a nova instância pertence. Clique em para exibir o projeto da empresa da instância.
	manutenção	Intervalo de tempo para quaisquer atividades de manutenção agendadas em nós de cache desta instância de DCS. Para modificar a janela, clique no ícone de .
	Descrição	Descrição da instância do DCS escolhida. Para modificar a descrição, clique no ícone de .
Conexão	Protegido por senha	Acesso protegido por senha ou sem senha.

Seção	Parâmetro	Descrição
	Endereço da conexão	Nome do domínio e número da porta da instância. Você pode clicar em ao lado de **Connection Address** para alterar a porta. **NOTA** ● Para uma instância principal/em espera do Redis 4.0/5.0, esse endereço indica o nome de domínio e o número da porta do nó principal. **Read-only Address** é o nome do domínio e o número da porta do nó em espera. Ao se conectar a uma instância desse tipo, você pode usar o nome de domínio e o número da porta do nó principal ou do nó em espera. ● Você pode alterar a porta apenas para uma instância do DCS Redis 4.0, 5.0 ou 6.0, mas não para uma instância do DCS Redis 3.0 ou do Memcached.
	Endereço IP	Endereço IP e número da porta da instância. O endereço de conexão do nome de domínio é recomendado.
	Acesso Público	Um indicador de que o acesso público está habilitado. **O acesso público é suportado apenas para o Redis 3.0, e não para 5.0, 4.0 e Memcached**.
	Endereço de acesso público	EIP vinculado à instância para acesso público. Esse parâmetro é exibido somente quando **Public Access** está habilitado. **NOTA** Clique em **Download Certificate for Public Access** para baixar um certificado, que pode ser usado para verificar o certificado da instância DCS quando você acessar a instância.
Rede	AZ	Zona de disponibilidade na qual residem os nós de cache que executam a instância de DCS selecionada.
	VPC	VPC na qual a instância escolhida reside.
	Sub-rede	Sub-rede na qual a instância escolhida reside.
	Grupo de segurança	Grupo de segurança que controla o acesso à instância escolhida. Para modificar o grupo de segurança, clique no ícone de . O controle de acesso do grupo de segurança é suportado apenas por instâncias do DCS Redis 3.0 e do Memcached. O DCS for Redis 4.0/5.0 é baseado no VPC Endpoint e não oferece suporte a grupos de segurança.
Topologia da instância	-	Passe o mouse sobre um nó para exibir suas métricas ou clique no ícone de um nó para exibir suas métricas históricas. As topologias são suportadas apenas para instâncias principal/em espera, Cluster de proxy e Cluster do Redis.
Cobrança	Modo de cobrança	Modo de cobrança da instância.
	Criação	Hora em que a instância escolhida começou a ser criada.

Seção	Parâmetro	Descrição
	Corra	Hora em que a instância foi criada.

----Fim

6.2 Modificando especificações

No console do DCS, você pode escalar uma instância do DCS Redis ou Memcached para uma capacidade maior ou menor, ou alterar o tipo de instância.

NOTA

- **Modificar especificações de instância durante horários fora de pico.** Se a modificação falhou em horários de pico (por exemplo, quando o uso da memória ou da CPU for superior a 90% ou quando houver picos de tráfego de gravação) Tente novamente fora do horário de pico.
- Se as instâncias do DCS forem muito antigas para suportar a modificação da especificação, entre em contato com o suporte técnico para fazer o upgrade das instâncias.
- O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.

Alteração do tipo de instância

Tabela 6-2 Opções de alteração do tipo de instância suportadas por diferentes instâncias de DCS

Versão	Alteração de tipo suportada	Precauções
Redis 3.0	De nó único para principal/em espera	A instância não pode ser conectada por vários segundos e permanece somente leitura por cerca de um minuto.
	Do principal/em espera para o cluster de proxy	1. Se os dados de uma instância principal/em espera do DCS Redis 3.0 estiverem armazenados em vários bancos de dados ou em bancos de dados não-DB0, a instância não poderá ser alterada para o tipo de cluster de proxy. Uma instância principal/em espera pode ser alterada para o tipo de cluster de proxy somente se seus dados forem armazenados somente no DB0. 2. A instância não pode ser conectada e permanece somente leitura por 5 a 30 minutos.
Memcached	De nó único para principal/em espera	Os serviços são interrompidos por vários segundos e permanecem somente leitura por cerca de 1 minuto.

Versão	Alteração de tipo suportada	Precauções
O Redis 4.0/5.0	Do principal/em espera para o cluster de proxy Da divisão de leitura/gravação ao Cluster de Proxy Do Cluster de Proxy para principal/em espera Do Cluster de Proxy para divisão de leitura/gravação	1. Antes de alterar o tipo de instância para Cluster de proxy, avalie o impacto nos serviços. Para obter detalhes, consulte **Quais são as restrições na implementação de vários bancos de dados em uma instância de cluster de proxy?** e **Restrições de Comando**. 2. O uso da memória deve ser inferior a 70% da memória máxima da nova variante. 3. Algumas chaves podem ser despejadas se o uso de memória atual exceder 90% do total. 4. Após a alteração, **criar regras de alarme** novamente para a instância. 5. Para instâncias que estão atualmente principal/em espera, certifique-se de que seu endereço IP ou nome de domínio somente leitura não seja usado pelo aplicativo. 6. Se o aplicativo não puder se reconectar ao Redis ou lidar com exceções, talvez seja necessário reiniciar o aplicativo após a alteração. 7. Modifique as especificações da instância fora dos horários de pico. Uma instância é temporariamente interrompida e permanece apenas para leitura por cerca de 1 minuto durante a alteração da especificação.

Quaisquer alterações de tipo de instância não listadas na tabela anterior não são suportadas. Para modificar especificações ao alterar o tipo de instância, consulte **Comutação de IP**.

Dimensionamento

- **Opções de dimensionamento**

Tabela 6-3 Opções de dimensionamento suportadas por instâncias diferentes

Mecanismo de cache	Único-nó	Principal/Em espera	Cluster do Redis	Cluster de proxy	Separação de leitura/gravação
O Redis 3.0	Escalando para cima/para baixo	Escalando para cima/para baixo	Escalando para cima/para baixo	Ampliando	-

Mecanismo de cache	Único-nó	Principal/Em espera	Cluster do Redis	Cluster de proxy	Separação de leitura/gravação
Redis 4.0	Escalando para cima/para baixo	Escalando para cima/para baixo, para fora/para dentro	Escalando para cima/para baixo, para fora/para dentro	Escalando para cima/para baixo	Escalando para cima/para baixo, para fora/para dentro
Redis 5.0	Escalando para cima/para baixo	Escalando para cima/para baixo, para fora/para dentro	Escalando para cima/para baixo, para fora/para dentro	Escalando para cima/para baixo	Escalando para cima/para baixo, para fora/para dentro
Memcached	Escalando para cima/para baixo	Escalando para cima/para baixo	-	-	-
versão básica do Redis 6.0	Escalando para cima/para baixo	Escalando para cima/para baixo	-	-	-
Edições profissionais do Redis 6.0	-	Nenhuma alteração é suportada.	-	-	-

NOTA

Se a memória reservada de uma instância do DCS Redis 3.0 ou Memcached for insuficiente, a modificação poderá falhar quando a memória for usada. Para obter detalhes, consulte **Memória Reservada**.

- **Impacto do escalonamento**

Tabela 6-4 Impacto do escalonamento

Tipos de instância	Tipo de dimensionamento	Impacto
Divisão de nó único, principal/em espera e leitura/gravação	Escalando para cima/para baixo	● Uma instância do DCS Redis 4.0 ou 5.0 será desconectada por vários segundos e permanecerá somente leitura por cerca de 1 minuto. Uma instância do DCS Redis 3.0 será desconectada e permanecerá somente leitura por 5 a 30 minutos. ● Para escalar, apenas a memória da instância é expandida. A capacidade de processamento da CPU não é melhorada. ● As instâncias de DCS de nó único não oferecem suporte à persistência de dados. Os dados não são retidos durante o dimensionamento. Após o dimensionamento, verifique se os dados estão completos e importe os dados, se necessário. Se houver dados importantes, use uma ferramenta de migração para migrar os dados para outras instâncias para backup. ● Os registros de backup de instâncias de divisão principal/em espera e de leitura/gravação não podem ser restaurados após a ampliação.

Tipos de instância	Tipo de dimensionamento	Impacto
Cluster de proxy e cluster do Redis	Escalando para cima/para baixo	● O dimensionamento envolve migração de dados, o que aumenta a latência de acesso. Para uma instância do Cluster do Redis, verifique se o cliente pode processar corretamente os comandos **MOVED** e **ASK**. Caso contrário, as solicitações falharão. ● Se a memória ficar cheia durante o escalonamento devido a uma grande quantidade de dados sendo gravados, o escalonamento falhará. ● Os registros de backup criados antes do dimensionamento não podem ser restaurados. ● Antes de dimensionar, verifique se há grandes chaves através da Análise de Cache. O Redis tem um limite na migração de chaves. Se a instância tiver uma única chave maior que 512 MB, o escalonamento falhará quando a migração de chave grande entre os nós expirar. Quanto maior a chave, maior a probabilidade de a migração falhar. ● Antes de aumentar ou diminuir a escala de uma instância do Redis Cluster, certifique-se de que a atualização automatizada da topologia do cluster esteja ativada se você usar o Lettuce. Se ele estiver desativado, você precisará reiniciar o cliente após o dimensionamento. Para obter detalhes sobre como ativar a atualização automatizada, consulte **um exemplo de uso do Lettuce para se conectar a uma instância do Redis Cluster**. ● A ampliação não interrompe as conexões, mas ocupa os recursos da CPU, diminuindo o desempenho em até 20%. ● Durante a expansão, novos nós do servidor Redis são adicionados e os dados são balanceados automaticamente para os novos nós. ● Para reduzir a escala de uma instância, certifique-se de que a memória usada de cada nó seja inferior a 70% da memória máxima por nó da nova variação. ● Se a quantidade de estilhaços diminuir durante a redução, os nós serão excluídos. Antes de reduzir a escala, certifique-se de que os nós excluídos não sejam referenciados diretamente no aplicativo, para evitar exceções de acesso ao serviço. ● Se a quantidade de estilhaços diminuir durante a redução, os nós serão excluídos e as conexões serão interrompidas. Se o aplicativo não puder se reconectar ao Redis ou lidar com exceções, talvez seja necessário reiniciá-lo após o escalonamento.

Tipos de instância	Tipo de dimensionamento	Impacto
Principal/em espera, divisão de leitura/gravação e instâncias de cluster do Redis	Escalando para fora/para dentro (alteração na quantidade de réplicas)	● Antes de escalar ou em uma instância de cluster do Redis, certifique-se de que a atualização automatizada da topologia do cluster esteja ativada se você usar o Lettuce. Se ele estiver desativado, você precisará reiniciar o cliente após o dimensionamento. Para obter detalhes sobre como ativar a atualização automatizada, consulte **um exemplo de uso do Lettuce para se conectar a uma instância do Redis Cluster**. ● A exclusão de réplicas interrompe as conexões. Se o aplicativo não puder se reconectar ao Redis ou lidar com exceções, será necessário reiniciar o aplicativo após o dimensionamento. ● Se o número de réplicas já for o mínimo suportado pela instância, você não poderá mais excluir réplicas.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Escolha **More** > **Modify Specifications** na linha que contém a ocorrência do DCS.

Passo 5 Na página **Modify Specifications**, selecione a especificação desejada.

NOTA

Se você optar por alterar a quantidade de réplicas de uma instância do Cluster do Redis, o campo **Added Replicas** será exibido na página. Modificar a especificação da instância fará com que a especificação da réplica seja modificada, o que resulta em uma alteração no preço.

Passo 6 Defina **Apply Change** para **Now** ou **During maintenance**.

Selecione **During maintenance** se a modificação interromper as conexões.

Tabela 6-5 Cenários em que a modificação da especificação interrompe as conexões

Alterar	Quando as conexões são interrompidas
Ampliação de uma instância de nó único ou principal/em espera	A memória é aumentada de um tamanho menor que 8 GB para 8 GB ou maior.

Alterar	Quando as conexões são interrompidas
Reduzir um Cluster de Proxy e uma instância de Cluster do Redis	O número de estilhaços é reduzido.
Alterando o tipo de instância	O tipo de instância é alterado entre divisão principal/em espera ou leitura/gravação e Cluster de proxy.
Exclusão de réplicas	As réplicas são excluídas de uma instância de divisão de leitura/gravação, cluster do Redis ou principal/em espera.

NOTA

- Se a modificação não interromper as conexões, ela será aplicada imediatamente, mesmo se você selecionar **During maintenance**.
- A modificação não pode ser retirada uma vez submetida. Para reprogramar uma modificação, você pode alterar a janela de manutenção. A janela de manutenção pode ser alterada até três vezes.
- As modificações em instâncias do DCS Redis 3.0 e do Memcached só podem ser aplicadas imediatamente.

Passo 7 Clique em **Next**, confirme os detalhes e clique em **Submit**.

Você pode ir para a página **Background Tasks** para exibir o status da modificação. Para obter mais informações, consulte **Exibindo Tarefas em Segundo Plano**.

A modificação da especificação de uma instância de DCS de nó único ou de principal/em espera leva de 5 a 30 minutos para ser concluída, enquanto a de uma instância de DCS de cluster leva mais tempo. Depois que uma instância é modificada com êxito, ela muda para o estado **Running**.

NOTA

- Se a modificação da especificação de uma instância de DCS de nó único falhar, a instância ficará temporariamente indisponível para uso. A especificação permanece inalterada. Algumas operações de gerenciamento (como configuração de parâmetros e modificação de especificações) não são suportadas temporariamente. Depois que a modificação da especificação é concluída no backend, a instância muda para a nova especificação e fica disponível para uso novamente.
- Se a modificação da especificação de uma instância de DCS principal/em espera ou de cluster falhar, a instância ainda estará disponível para uso com suas especificações originais. Algumas operações de gerenciamento (como configuração de parâmetros, backup, restauração e modificação de especificação) não são suportadas temporariamente. Lembre-se de não ler ou gravar mais dados do que o permitido pelas especificações originais; caso contrário, pode ocorrer perda de dados.
- Depois que a modificação da especificação for bem-sucedida, a nova especificação da instância entrará em vigor.

----Fim

6.3 Iniciando uma instância

No console do DCS, você pode iniciar uma ou várias instâncias do DCS por vez.

Quando uma instância de cluster é iniciada, o status e os dados são sincronizados entre os nós da instância. Se uma grande quantidade de dados for gravada continuamente na instância

antes da conclusão da sincronização, a sincronização será prolongada e a instância permanecerá no estado **Starting**. Após a conclusão da sincronização, a instância entra no estado **Running**.

NOTA

Esta função é suportada apenas por instâncias antigas do DCS Redis no estado **Stopped**. Novas instâncias não podem ser iniciadas ou interrompidas.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Selecione a instância que deseja iniciar e clique em **Start** acima da lista de instâncias do DCS.

Passo 5 Na caixa de diálogo exibida, clique em **Yes**.

- Leva de 1 a 30 minutos para iniciar instâncias de DCS.
- Depois que as instâncias DCS são iniciadas, seus status mudam de **Stopped** para **Running**.

NOTA

Para iniciar uma única instância, clique em **Start** na coluna **Operation** na linha que contém a instância desejada.

----Fim

6.4 Reiniciando uma Instância

No console do DCS, você pode reiniciar uma ou várias instâncias do DCS por vez.

ATENÇÃO

- Depois que uma instância de DCS de nó único for reiniciada, os dados serão excluídos da instância.
- Enquanto uma instância de DCS estiver reiniciando, ela não poderá ser lida ou gravada.
- Uma tentativa de reiniciar uma instância de DCS durante o backup pode resultar em uma falha.

Pré-requisitos

As instâncias de DCS que você deseja reiniciar estão no estado **Running** ou **Faulty**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, selecione uma ou mais instâncias de DCS que deseja reiniciar.

Passo 5 Clique em **Restart** acima da lista de instâncias do DCS.

Passo 6 Na caixa de diálogo exibida, clique em **Yes**.

Leva de **1 to 30** minutos para reiniciar instâncias de DCS. Depois que as instâncias de DCS são reiniciadas, seu status muda para **Running**.

NOTA

- Por padrão, apenas o processo da instância será reiniciado. Se você selecionar **Force restart** para uma instância do DCS Redis 3.0 ou do Memcached, a VM será reiniciada. **Force restart** não é suportada pelas instâncias do DCS Redis 4.0 ou posteriores.
- Para reiniciar uma única instância, você também pode clicar em **Restart** na linha que contém a instância desejada.
- O tempo necessário para reiniciar uma instância de DCS depende do tamanho do cache da instância.

----Fim

6.5 Deletando uma Instância

No console do DCS, você pode excluir uma ou várias instâncias do DCS por vez. Você também pode excluir todas as tarefas de criação de instâncias que falharam na execução.

AVISO

- Depois que uma instância DCS for excluída, os dados da instância serão excluídos sem backup. Além disso, todos os dados de backup da instância serão excluídos. Portanto, baixe os arquivos de backup da instância para armazenamento permanente antes de excluir a instância.
- Se a instância estiver no modo de cluster, todos os nós do cluster serão excluídos.
- As instâncias cobradas anualmente/mensalmente não podem ser excluídas.

Pré-requisitos

- As instâncias de DCS que você deseja excluir foram criadas.
- As instâncias de DCS que você deseja excluir estão no estado **Running**, **Faulty**, ou **Stopped**.

Procedimento

Deletando Instâncias de DCS

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, selecione uma ou mais instâncias de DCS que deseja deletar.

As instâncias de DCS no estado **Creating**, **Starting**, **Stopping**, ou **Restarting** não podem ser excluídas.

Passo 5 Escolha **More** > **Delete** acima da lista de instâncias.

Passo 6 Digite **DELETE** e clique em **Yes** para excluir a instância do DCS.

Leva de 1 a 30 minutos para excluir instâncias de DCS.

NOTA

Para excluir uma única instância, escolha a coluna **More** > **Delete** na **Operation** na linha que contém a instância.

----Fim

Deletando Tarefas de Criação de Instância que Falharam ao Executar

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região e um projeto.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Se houver instâncias de DCS que falharam ao serem criadas, **Instance Creation Failures** e o número de instâncias que falham ao serem criadas serão exibidos acima da lista de instâncias.

Passo 4 Clique no ícone ou no número de tarefas que falharam ao lado de **Instance Creation Failures**.

A caixa de diálogo **Instance Creation Failures** é exibida.

Passo 5 Exclua as tarefas de criação de instâncias com falha, conforme necessário.

- Para excluir todas as tarefas com falha, clique em **Delete All** acima da lista de tarefas.
- Para excluir uma única tarefa com falha, clique em **Delete** na linha que contém a tarefa.

----Fim

6.6 Executando um switchover principal/em espera

No console de DCS, você pode alternar manualmente os nós principais e em espera de uma instância de DCS principal/em espera. Essa operação é usada para fins especiais, por exemplo, liberar todas as conexões de serviço ou encerrar as operações de serviço em andamento.

Para executar uma alternância manual para uma instância de Cluster de Proxy ou Cluster do Redis DCS Redis 4.0 ou 5.0, acesse a página **Shards and Replicas** da instância. Para mais detalhes, consulte **Gerenciando Fragmentos e Réplicas**.

AVISO

- Durante a alternância principal/em espera, os serviços serão interrompidos por até 10 segundos. Antes de executar essa operação, verifique se o aplicativo suporta o restabelecimento da conexão em caso de desconexão.
- Durante uma alternância de nó principal/em espera, uma grande quantidade de recursos será consumida para sincronização de dados entre os nós principais e nós em espera. É aconselhável realizar esta operação fora do horário de pico.
- Os dados dos nós principais e em espera são sincronizados de forma assíncrona. Portanto, uma pequena quantidade de dados que está sendo operada durante o switchover pode ser perdida.

Pré-requisitos

A instância DCS para a qual você deseja executar uma alternância principal/em espera está no estado **Running**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na coluna **Operation** da instância, escolha **More** > **Master/Standby Switchover**.

----Fim

6.7 Limpando dados de instância do DCS

No console do DCS, você pode limpar dados somente para instâncias do DCS Redis 4.0/5.0. A limpeza dos dados da instância não pode ser desfeita e os dados apagados não podem ser recuperados. Tenha cuidado ao realizar esta operação.

Pré-requisitos

A instância do DCS Redis 4.0/5.0 está no estado **Running**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Selecione uma ou mais instâncias de DCS.

Passo 5 Escolha **More** > **Clear** acima da lista de instâncias.

Passo 6 Na caixa de diálogo exibida, clique em **Yes**.

----Fim

6.8 Exportando Lista de Instâncias

No console do DCS, você pode exportar as informações completas da instância do DCS para um arquivo do Excel.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique em  acima da lista de instâncias.

Clique no resultado de exportação exibido no canto inferior esquerdo da página. **Figura 6-1** mostra um resultado de exemplo.

Figura 6-1 Lista de instâncias de DCS exportadas

Name	ID	Status	AZ	Cache Eng	Instance	Specifica	Used/Avai	Connecti	Created	Billing M	VPC	VPC ID	Enterprise Project
dcs-trpt	5e4f4c58-	Running	AZ1	Redis 5.(	Single-no	0.125	0/128 (0%	198.19.32	May 24, 2	Free	null	null	default
dcs-APITe	693491b0-	Running		Redis 3.(	Master/St	2	2/1,536 (	172.16.14	May 06, 2	Yearly/Mo	null	52267da0-	default

----Fim

6.9 Renomeando comandos

Depois de criar uma instância do DCS Redis 4.0/5.0, você pode renomear os seguintes comandos críticos: Atualmente, você só pode renomear os comandos **COMMAND**, **KEYS**, **FLUSHDB**, **FLUSHALL**, e **HGETALL**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na coluna **Operation** de uma instância, escolha **More** > **Command Renaming**.

Passo 5 Selecione um comando, insira um novo nome e clique em **OK**.

NOTA

- Você pode renomear vários comandos ao mesmo tempo.
- Os novos nomes de comandos só terão efeito depois de reiniciar a instância. Lembre-se dos novos nomes de comando porque eles não serão exibidos no console por motivos de segurança.

----Fim

7 Gerenciando instâncias de DCS

7.1 Aviso de configuração

Na maioria dos casos, diferentes operações de gerenciamento de instâncias de DCS não podem prosseguir simultaneamente. Se você iniciar uma nova operação de gerenciamento enquanto a operação atual estiver em andamento, o console de DCS solicitará que você inicie a nova operação novamente após a conclusão da operação atual. As operações de gerenciamento de instâncias de DCS incluem:

- Criando uma instância de DCS
- Configurando parâmetros
- Reiniciando uma instância de DCS
- Alteração da senha da instância
- Redefinindo a senha da instância
- Dimensionamento, backup ou restauração de uma instância

Você pode reiniciar uma instância de DCS durante o backup, mas a tarefa de backup será forçadamente interrompida e provavelmente resultará em uma falha de backup.

AVISO

Caso um nó de cache de uma instância do DCS esteja com defeito:

- A instância permanece no estado **Running** e você pode continuar lendo e gravando a instância. Isto é conseguido graças à alta disponibilidade do DCS.
- Os nós de cache podem se recuperar de falhas internas automaticamente. A recuperação manual de falhas também é suportada.
- Certas operações (como backup, restauração e configuração de parâmetros) na zona de gerenciamento não são suportadas durante a recuperação de falhas. Você pode entrar em contato com o atendimento ao cliente ou executar essas operações depois que os nós de cache se recuperam de falhas.

7.2 Modificando Parâmetros de Configuração

7.2.1 Modificando Parâmetros de Configuração de uma Instância

No console do DCS, você pode configurar parâmetros para uma instância para obter o desempenho ideal do DCS.

Por exemplo, se você não precisar de persistência de dados, defina **appendonly** como **no**.

Depois que os parâmetros de configuração da instância são modificados, a modificação entra em vigor imediatamente sem a necessidade de reiniciar manualmente a instância. Para uma instância de cluster, a modificação entra em vigor em todos os estilhaços.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, clique no nome da instância do DCS que deseja configurar.

Passo 5 Na página de detalhes da instância, escolha **Instance Configuration** > **Parameters**.

Passo 6 Clique em **Modify**.

Passo 7 Modifique os parâmetros com base nas suas necessidades.

Tabela 7-1 e **Tabela 7-2** descrever os parâmetros. Na maioria dos casos, os valores padrão são mantidos.

Tabela 7-1 Parâmetros de configuração da instância do DCS Redis

Parameter	Descrição	Value Range	Default Value
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0

Parameter	Descrição	Value Range	Default Value
apêndicefsync	Controla a frequência com que o fsync() transfere dados em cache para o disco. Observe que alguns SO realizarão uma transferência de dados completa, mas alguns outros apenas fazem uma tentativa de "melhor esforço". Existem três configurações: no: fsync() nunca é chamado. O SO liberará os dados quando estiver pronto. Este modo oferece o mais alto desempenho. sempre: fsync() é chamado após cada gravação no AOF. Este modo é muito lento, mas também muito seguro. Everysec: fsync() é chamado uma vez por segundo. Este modo proporciona um compromisso entre segurança e desempenho.	● não ● Sempre ● a cada seg	a cada seg
apenas anexação	Indica se cada modificação da instância deve ou não ser registrada. Por padrão, dados são gravados em discos de maneira assíncrona no Redis. Se essa função estiver desativada, os dados gerados recentemente poderão ser perdidos no caso de uma falha de energia. Opções: **yes**: Os logs são ativados, ou seja, a persistência é ativada. **no**: Os logs são desabilitados, ou seja, a persistência é desabilitada.	● Sim ● não	Sim

Parameter	Descrição	Value Range	Default Value
client-output-buffer-limit-slave-soft-segundos	Número de segundos que o buffer de saída permanece acima do **client-output-buffer-slave-soft-limit** antes que o cliente seja desconectado.	0–60	60
client-output-buffer-slave-hard-limit	Limite rígido (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite rígido, o cliente é imediatamente desconectado.	0–17.179.869.184	1.717.986.918
client-output-buffer-slave-soft-limit	Limite suave (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite suave e permanece continuamente acima do limite pelo tempo especificado pelo parâmetro **client-output-buffer-limit-slave-soft-seconds**, o cliente é desconectado.	0–17.179.869.184	1.717.986.918

Parameter	Descrição	Value Range	Default Value
política de maxmemória	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	● volatile-lru ● allkeys-lru ● volatile-random ● allkeys-random ● volatile-ttl ● noeviction	volatile-lru **NOTA** Se a instância do DCS Redis for criada antes de julho de 2020 e esse parâmetro não tiver sido modificado, o valor padrão será **noeviction**. Se a instância for criada depois de julho de 2020, o valor padrão será **volatile-lru**.
lua-limite de tempo	Tempo máximo permitido para executar um script Lua (em milissegundos).	100–5000	5000
mestre-somente-leitura	Define a instância como somente leitura. Todas as operações de escrita falharão.	● Sim ● não	não
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–50.000	10.000
proto-max-bulk-len	Tamanho máximo de uma solicitação de um único elemento (em bytes).	1.048.576–536.870.912	536.870.912

Parameter	Descrição	Value Range	Default Value
repl-backlog-tamanho	O tamanho do backlog de replicação (bytes). O backlog é um buffer que acumula dados de réplica quando réplicas são desconectadas do principal. Quando uma réplica é reconectada, uma sincronização parcial é realizada para sincronizar os dados que foram perdidos enquanto as réplicas eram desconectadas.	16.384–1.073.741.824	1.048.576
repl-backlog-ttl	A quantidade de tempo, em segundos, antes do buffer de backlog ser liberado, a partir da última vez que uma réplica foi desconectada. O valor **0** indica que o backlog nunca é liberado.	0–604.800	3600
repl- timeout	Tempo limite de replicação (em segundos).	30–3600	60
hash-max-ziplist-entradas	O número máximo de hashes que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	512
hash-max-ziplist-value	O maior valor permitido para um hash codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
set-max-intset-entradas	Se um conjunto é composto inteiramente de cadeias de caracteres que são inteiros em radix 10 dentro do intervalo de inteiros com sinal de 64 bits, o conjunto é codificado usando intset, uma estrutura de dados otimizada para uso de memória.	1–10.000	512

Parameter	Descrição	Value Range	Default Value
zset-max-ziplist-entradas	O número máximo de conjuntos classificados que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	128
zset-max-ziplist-valor	O maior valor permitido para um conjunto ordenado codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
latência-monitor-limiar	A quantidade mínima de latência que será registrada como picos de latência ● configure para 0: O monitoramento de latência está desativado. ● Definir como mais de 0: Tudo com pelo menos este tempo de latência (em ms) será registrado. Ao executar o comando **LATENCY**, você pode executar operações relacionadas ao monitoramento de latência, como obter dados estatísticos e configurar e ativar o monitoramento de latência.	0–86.400.000 ms	0

Parameter	Descrição	Value Range	Default Value
notific-keyspace-events	Controla para quais notificações de eventos de keyspace são ativadas. Se esse parâmetro estiver configurado, o recurso Redis Pub/Sub permitirá que os clientes recebam uma notificação de evento quando um conjunto de dados do Redis for modificado. As instâncias de cluster de proxy não têm esse parâmetro.	Uma combinação de valores diferentes pode ser usada para ativar notificações para vários tipos de eventos. Os valores possíveis incluem: K: Eventos de espaço de chave, publicados com o prefixo __keyspace@__ e: Eventos keyevent, publicados com o prefixo __keyevent@__ g: Comandos genéricos (não específicos do tipo), como DEL, EXPIRE e RENAME $: Comandos de string eu: Comandos de lista s: Definir comandos h: Comandos de hash z: Comandos do conjunto classificado x: Eventos expirados (eventos gerados toda vez que uma chave expira) e: Eventos despejados (eventos gerados quando uma chave é despejada da maxmemory) Para obter mais informações, consulte a seguinte nota.	Ex
slowlog-log-mais lento-do que	A quantidade máxima de tempo permitida, em microssegundos, para execução de comandos. Se esse limite for excedido, o log de consultas lentas do Redis registrará o comando.	0–1.000.000	10.000

Parameter	Descrição	Value Range	Default Value
slowlog-max-len	O número máximo permitido de consultas lentas que podem ser registradas. O log de consulta lento consome memória, mas você pode recuperar essa memória executando o comando **SLOWLOG RESET**.	0–1000	128

NOTA

1. Para obter mais informações sobre os parâmetros descritos em **Tabela 7-1**, visite **https://redis.io/topics/memory-optimization**.
2. O parâmetro **latency-monitor-threshold** é normalmente usado para localização de falhas. Depois de localizar falhas com base nas informações de latência coletadas, altere o valor de **latency-monitor-threshold** para **0** para evitar latência desnecessária.
3. Mais informações sobre o parâmetro **notify-keyspace-events**:
   - A configuração do parâmetro deve conter pelo menos um **K** ou **E**.
   - **A** é um apelido para "g$lshzxe" e não pode ser usado junto com qualquer um dos caracteres em "g$lshzxe".
   - Por exemplo, o valor **Kl** significa que o Redis pode notificar clientes Pub/Sub sobre eventos de espaço de chaves e comandos de lista. O valor **AKE** significa que o Redis notificará os clientes do Pub/Sub sobre todos os eventos.
4. Os parâmetros configuráveis variam dependendo do tipo de instância.

Tabela 7-2 Parâmetros de configuração da instância do Memcached DCS

Parâmetro	Descrição	Intervalo de valores	Valor padrão
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–10.000	10.000

Parâmetro	Descrição	Intervalo de valores	Valor padrão
maxmemory-policy	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	volátil-lru allkeys-lru volátil-aleatório allkeys-aleatório volátil-ttl noeviction	noeviction
reservada-memória-porcentagem	Percentual da memória máxima disponível reservada para processos em segundo plano, como persistência e replicação de dados.	0–80	30

Passo 8 Depois de terminar de definir os parâmetros, clique em **Save**.

Passo 9 Clique em **Yes** para confirmar a modificação.

----Fim

7.2.2 Modificando Parâmetros de Configuração em Lotes

No console do DCS, você pode configurar vários parâmetros por vez para que uma instância atinja o desempenho ideal do DCS.

Depois que os parâmetros de configuração da instância são modificados, a modificação entra em vigor imediatamente sem a necessidade de reiniciar manualmente a instância. Para uma instância de cluster, a modificação entra em vigor em todos os estilhaços.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na página **Cache Manager**, selecione todas as instâncias de DCS que deseja configurar.

Passo 5 Escolha **More** > **Modify Parameters**.

NOTA

- Os parâmetros exibidos na página **Modify Parameters** são o conjunto de uniões dos parâmetros das instâncias selecionadas. Por exemplo, suponha que o parâmetro **appendfsync** da instância 1 não é suportado pela instância 2. Você ainda pode selecionar esse parâmetro, mas o sistema exibe uma mensagem em **Passo 7** indicando que a instância 2 não oferece suporte a esse parâmetro. Após o envio, o comando de modificação não será entregue à instância 2. O intervalo de valores do parâmetro é a interseção dos intervalos de valores do parâmetro das instâncias selecionadas. Por exemplo, se o intervalo de valores for de 0 a 50.000, por exemplo, 1, e de 1000 a 50.000, por exemplo, 2, o intervalo de valores exibido na página será de 1000 a 50.000.
- Se não quiser modificar uma instância selecionada, clique em ao lado dela.

Passo 6 Selecione o parâmetro a ser modificado e insira um novo valor na coluna **New Value**.

Tabela 7-3 e **Tabela 7-4** descrevem os parâmetros. Na maioria dos casos, você pode manter os valores padrão.

Tabela 7-3 Parâmetros de configuração da instância do DCS Redis

Parameter	Descrição	Value Range	Default Value
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0

Parameter	Descrição	Value Range	Default Value
apêndicefsync	Controla a frequência com que o fsync() transfere dados em cache para o disco. Observe que alguns SO realizarão uma transferência de dados completa, mas alguns outros apenas fazem uma tentativa de "melhor esforço". Existem três configurações: no: fsync() nunca é chamado. O SO liberará os dados quando estiver pronto. Este modo oferece o mais alto desempenho. sempre: fsync() é chamado após cada gravação no AOF. Este modo é muito lento, mas também muito seguro. Everysec: fsync() é chamado uma vez por segundo. Este modo proporciona um compromisso entre segurança e desempenho.	● não ● Sempre ● a cada seg	a cada seg
apenas anexação	Indica se cada modificação da instância deve ou não ser registrada. Por padrão, dados são gravados em discos de maneira assíncrona no Redis. Se essa função estiver desativada, os dados gerados recentemente poderão ser perdidos no caso de uma falha de energia. Opções: **yes**: Os logs são ativados, ou seja, a persistência é ativada. **no**: Os logs são desabilitados, ou seja, a persistência é desabilitada.	● Sim ● não	Sim

Parameter	Descrição	Value Range	Default Value
client-output-buffer-limit-slave-soft-segundos	Número de segundos que o buffer de saída permanece acima do **client-output-buffer-slave-soft-limit** antes que o cliente seja desconectado.	0–60	60
client-output-buffer-slave-hard-limit	Limite rígido (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite rígido, o cliente é imediatamente desconectado.	0–17.179.869.184	1.717.986.918
client-output-buffer-slave-soft-limit	Limite suave (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite suave e permanece continuamente acima do limite pelo tempo especificado pelo parâmetro **client-output-buffer-limit-slave-soft-seconds**, o cliente é desconectado.	0–17.179.869.184	1.717.986.918

Parameter	Descrição	Value Range	Default Value
política de maxmemória	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	● volatile-lru ● allkeys-lru ● volatile-random ● allkeys-random ● volatile-ttl ● noeviction	volatile-lru **NOTA** Se a instância do DCS Redis for criada antes de julho de 2020 e esse parâmetro não tiver sido modificado, o valor padrão será **noeviction**. Se a instância for criada depois de julho de 2020, o valor padrão será **volatile-lru**.
lua-limite de tempo	Tempo máximo permitido para executar um script Lua (em milissegundos).	100–5000	5000
mestre-somente-leitura	Define a instância como somente leitura. Todas as operações de escrita falharão.	● Sim ● não	não
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–50.000	10.000
proto-max-bulk-len	Tamanho máximo de uma solicitação de um único elemento (em bytes).	1.048.576–536.870.912	536.870.912

Parameter	Descrição	Value Range	Default Value
repl-backlog-tamanho	O tamanho do backlog de replicação (bytes). O backlog é um buffer que acumula dados de réplica quando réplicas são desconectadas do principal. Quando uma réplica é reconectada, uma sincronização parcial é realizada para sincronizar os dados que foram perdidos enquanto as réplicas eram desconectadas.	16.384–1.073.741.824	1.048.576
repl-backlog-ttl	A quantidade de tempo, em segundos, antes do buffer de backlog ser liberado, a partir da última vez que uma réplica foi desconectada. O valor **0** indica que o backlog nunca é liberado.	0–604.800	3600
repl- timeout	Tempo limite de replicação (em segundos).	30–3600	60
hash-max-ziplist-entradas	O número máximo de hashes que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	512
hash-max-ziplist-value	O maior valor permitido para um hash codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
set-max-intset-entradas	Se um conjunto é composto inteiramente de cadeias de caracteres que são inteiros em radix 10 dentro do intervalo de inteiros com sinal de 64 bits, o conjunto é codificado usando intset, uma estrutura de dados otimizada para uso de memória.	1–10.000	512

Parameter	Descrição	Value Range	Default Value
zset-max-ziplist-entradas	O número máximo de conjuntos classificados que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	128
zset-max-ziplist-valor	O maior valor permitido para um conjunto ordenado codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
latência-monitor-limiar	A quantidade mínima de latência que será registrada como picos de latência ● configure para 0: O monitoramento de latência está desativado. ● Definir como mais de 0: Tudo com pelo menos este tempo de latência (em ms) será registrado. Ao executar o comando **LATENCY**, você pode executar operações relacionadas ao monitoramento de latência, como obter dados estatísticos e configurar e ativar o monitoramento de latência.	0–86.400.000 ms	0

Parameter	Descrição	Value Range	Default Value
notific-keyspace-events	Controla para quais notificações de eventos de keyspace são ativadas. Se esse parâmetro estiver configurado, o recurso Redis Pub/Sub permitirá que os clientes recebam uma notificação de evento quando um conjunto de dados do Redis for modificado. As instâncias de cluster de proxy não têm esse parâmetro.	Uma combinação de valores diferentes pode ser usada para ativar notificações para vários tipos de eventos. Os valores possíveis incluem: K: Eventos de espaço de chave, publicados com o prefixo __keyspace@__ e: Eventos keyevent, publicados com o prefixo __keyevent@__ g: Comandos genéricos (não específicos do tipo), como DEL, EXPIRE e RENAME $: Comandos de string eu: Comandos de lista s: Definir comandos h: Comandos de hash z: Comandos do conjunto classificado x: Eventos expirados (eventos gerados toda vez que uma chave expira) e: Eventos despejados (eventos gerados quando uma chave é despejada da maxmemory) Para obter mais informações, consulte a seguinte nota.	Ex
slowlog-log-mais lento-do que	A quantidade máxima de tempo permitida, em microssegundos, para execução de comandos. Se esse limite for excedido, o log de consultas lentas do Redis registrará o comando.	0–1.000.000	10.000

Parameter	Descrição	Value Range	Default Value
slowlog-max-len	O número máximo permitido de consultas lentas que podem ser registradas. O log de consulta lento consome memória, mas você pode recuperar essa memória executando o comando **SLOWLOG RESET**.	0–1000	128

NOTA

1. Para obter mais informações sobre os parâmetros descritos em **Tabela 7-3**, visite o **site oficial do Redis**.
2. O parâmetro **latency-monitor-threshold** é normalmente usado para localização de falhas. Depois de localizar falhas com base nas informações de latência coletadas, altere o valor de **latency-monitor-threshold** para **0** para evitar latência desnecessária.
3. Mais informações sobre o parâmetro **notify-keyspace-events**:
   - A configuração do parâmetro deve conter pelo menos um **K** ou **E**.
   - **A** é um apelido para "g$lshzxe" e não pode ser usado junto com qualquer um dos caracteres em "g$lshzxe".
   - Por exemplo, o valor **Kl** significa que o Redis pode notificar clientes Pub/Sub sobre eventos de espaço de chaves e comandos de lista. O valor **AKE** significa que o Redis notificará os clientes do Pub/Sub sobre todos os eventos.

Tabela 7-4 Parâmetros de configuração da instância do Memcached DCS

Parâmetro	Descrição	Intervalo de valores	Valor padrão
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–10.000	10.000

Parâmetro	Descrição	Intervalo de valores	Valor padrão
maxmemory-policy	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	volátil-lru allkeys-lru volátil-aleatório allkeys-aleatório volátil-ttl noeviction	noeviction
reservada-memória-porcentagem	Percentual da memória máxima disponível reservada para processos em segundo plano, como persistência e replicação de dados.	0–80	30

Passo 7 Clique em **Next: Confirm Parameters** para confirmar as instâncias e os valores dos parâmetros.

NOTA

- Na página de confirmação, você pode filtrar as instâncias desejadas selecionadas em **Passo 4** por mecanismo de cache, tipo de instância e status e modificar o valor do parâmetro dessas instâncias.
- Você pode definir valores diferentes para instâncias diferentes dentro do intervalo de valores. Se o valor atual e o novo valor de uma instância forem iguais, nenhum registro de modificação será gerado para a instância.

Passo 8 Clique em **Submit**.

Passo 9 Clique em uma instância de DCS. Na página de detalhes da instância exibida, escolha **Parameters** > **Modification History** para verificar se o parâmetro de configuração foi modificado com êxito.

----Fim

7.3 Modificando a Janela Manutenção

No console do DCS, depois de criar uma instância do DCS, você pode modificar a janela de manutenção da instância do DCS na página **Basic Information** da instância. Durante a janela de manutenção, o pessoal de O&M pode manter a instância.

Pré-requisitos

Uma instância de DCS foi criada.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome da instância DCS desejada.

Passo 5 Clique na guia **Basic Information**. Na área **Instance Details**, clique no ícone de ao lado do parâmetro **Maintenance**.

Passo 6 Selecione uma nova janela de manutenção na lista suspensa. Clique em para salvar a modificação ou em para descartar a modificação.

A modificação entrará em vigor imediatamente na página de guia **Basic Information**.

----Fim

7.4 Modificando o Grupo de Segurança

No console do DCS, depois de criar uma instância do DCS, você pode modificar o grupo de segurança da instância do DCS na página **Basic Information** da instância.

Você pode modificar os grupos de segurança das instâncias do DCS Redis 3.0, mas não os das instâncias do DCS Redis 4.0/5.0.

NOTA

O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.

Pré-requisitos

Uma instância de DCS foi criada.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome da instância DCS desejada.

Passo 5 Clique na guia **Basic Information**. Na área **Network**, clique em junto ao parâmetro **Security Group**.

Passo 6 Selecione um novo grupo de segurança na lista suspensa. Clique em para salvar a modificação ou em para descartar a modificação.

NOTA

Somente os grupos de segurança que foram criados podem ser selecionados na lista suspensa. Se você precisar criar um grupo de segurança, siga o procedimento descrito em **Como configurar um grupo de segurança?**

A modificação entrará em vigor imediatamente na página de guia **Basic Information**.

----Fim

7.5 Exibindo Tarefas em Segundo Plano

Depois de iniciar determinadas operações de instância, como dimensionar a instância e alterar ou redefinir uma senha, uma tarefa em segundo plano será iniciada para cada operação. No console DCS, você pode exibir o status da tarefa em segundo plano e limpar as informações da tarefa excluindo os registros da tarefa.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Filtre instâncias de DCS para localizar a instância de DCS desejada. Atualmente, você pode pesquisar instâncias por nome, especificação, ID, endereço IP, AZ, status, tipo de instância, mecanismo de cache e muitos outros atributos.

Passo 4 Clique no nome da instância do DCS para exibir mais detalhes sobre a instância do DCS.

Passo 5 Clique na guia **Background Tasks**.

Uma lista de tarefas em segundo plano é exibida.

Passo 6 Clique em  , especifique **Start Date** e **End Date**, e clique em **OK** para exibir as tarefas iniciadas no segmento de tempo correspondente.

- Clique em  para atualizar o status da tarefa.
- Para limpar o registro de uma tarefa em segundo plano, escolha **Operation** > **Delete**.

  NOTA

  Você só pode excluir os registros de tarefas no estado **Successful** ou **Failed**.

----Fim

7.6 Gerenciando a lista branca de endereço IP

As instâncias do DCS Redis 3.0/4.0/5.0 e do Memcached são implantadas em modos diferentes. Portanto, o método de controle de acesso varia.

- Para controlar o acesso às instâncias do DCS Redis 3.0, Memcached e Redis 6.0 Professional Edition, você pode usar grupos de segurança. Whitelists não são suportadas. Para obter detalhes sobre como configurar um grupo de segurança, consulte **Como configurar um grupo de segurança?**
- Para controlar o acesso às instâncias do DCS Redis 4.0/5.0, você pode usar listas de permissões. Grupos de segurança não são suportados.

A seguir, descrevemos como gerenciar as listas de permissão de uma instância do Redis 4.0/5.0 para permitir o acesso somente de endereços IP da lista de permissão. Se nenhuma

lista branca for adicionada para a instância ou se a função de lista branca for desativada, todos os endereços IP que podem se comunicar com a VPC poderão acessar a instância.

Criando um grupo de whitelist

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome de uma instância de DCS.

Passo 5 Escolha **Instance Configuration** > **Whitelist**. Na página exibida, clique em **Create Whitelist Group**.

Passo 6 Na caixa de diálogo **Create Whitelist Group**, especifique **Group Name** e **IP Address/Range**.

Tabela 7-5 Parâmetros da lista branca

Parâmetro	Descrição	Exemplo
Nome de grupo	Nome do grupo Whitelist da instância. Um máximo de quatro grupos de listas brancas podem ser criados para cada instância.	DCS-teste
Endereço/intervalo de IP	Um máximo de 20 endereços IP ou intervalos de endereços IP podem ser adicionados a uma instância. Separe vários endereços IP ou intervalos de endereços IP com vírgulas. Endereço IP e intervalo de endereços IP não suportados: 0.0.0.0 e 0.0.0/0.	10.10.10.1,10.10.10.10

Passo 7 Clique em **OK**.

Um grupo de lista branca é ativado automaticamente para a instância uma vez criada. Somente endereços IP na lista de permissões podem acessar a instância.

NOTA

- Na lista de grupos da lista branca, clique em **Edit** para modificar os endereços IP ou intervalos de endereços IP em um grupo e clique em **Delete** para excluir um grupo da lista branca.
- Depois que a lista branca for ativada, você poderá clicar em **Disable Whitelist** acima da lista de grupos da lista branca para permitir que todos os endereços IP conectados à VPC acessem a instância.

----Fim

7.7 Gerenciando Tags

As tags facilitam a identificação e o gerenciamento de instâncias do DCS.

Você pode adicionar tags a uma instância ao criá-la ou adicionar, modificar ou excluir tags na página de detalhes de uma instância criada. Cada instância pode ter no máximo 20 tags.

Uma tag consiste em uma chave de tag e um valor de tag. **Tabela 7-6** lista os requisitos de chave e valor da tag.

Tabela 7-6 Requisitos de chave e valor da etiqueta

Parâmetro	Exigências
Chave da tag	● Não pode ser deixado em branco. ● Deve ser exclusivo para a mesma instância. ● Consiste em um máximo de 128 caracteres. ● Pode conter letras de qualquer idioma, dígitos, espaços e caracteres especiais _ . : = + - @ ● Não é possível iniciar ou terminar com um espaço. ● Não é possível iniciar com **_sys_**.
Valor da tag	● Consiste em um máximo de 255 caracteres. ● Pode conter letras de qualquer idioma, dígitos, espaços e caracteres especiais _ . : / = + - @ ● Não é possível iniciar ou terminar com um espaço.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

> NOTA
>
> Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome da instância do DCS desejada para ir para a página de detalhes.

Passo 5 Escolha **Instance Configuration** > **Tags**.

Passo 6 Execute as seguintes operações conforme necessário:

- Adicione uma tag.
  a. Clique em **Add/Edit Tag**.
     Se você tiver criado tags predefinidas, selecione um par predefinido de chave e valor de tag. Para exibir ou criar tags predefinidas, clique em **View predefined tags**. Em seguida, você será direcionado para o console do TMS.
     Você também pode criar novas tags especificando **Tag key** e **Tag value**.
  b. Clique em **OK**.
- Modificar uma etiqueta
  Clique em **Add/Edit Tag**. Na caixa de diálogo **Add/Edit Tag** exibida, exclua a chave desejada, adicione a chave novamente, insira um novo valor de tag e clique em **Add**.
- Eliminar uma etiqueta
  Na linha que contém a tag desejada, clique em **Delete**. Na caixa de diálogo exibida, clique em **Yes**.

----Fim

7.8 Gerenciando Fragmentos e Réplicas

Esta seção descreve como consultar os fragmentos e réplicas de uma instância principal/em espera, cluster, ou de leitura/gravação dividida do DCS Redis 4.0/5.0, e como promover manualmente uma réplica para principal.

As instâncias do DCS Redis 3.0 e as instâncias do DCS Redis 4.0/5.0/6.0 de nó único não suportam essa função.

- Por padrão, uma instância de divisão principal/em espera ou de leitura/gravação tem apenas um fragmento com um principal e uma réplica. Você pode exibir as informações de sharding na página **Shards and Replicas**. Para alternar manualmente as funções principais e réplicas, consulte **Executando um switchover principal/em espera**.
- Um Cluster de Proxy ou uma instância de Cluster do Redis tem vários estilhaços. Cada fragmento tem um principal e uma réplica. Na página **Shards and Replicas**, você pode exibir as informações de fragmento e alternar manualmente as funções principais e réplicas.

  > NOTA
  >
  > - Para obter detalhes sobre o número de estilhaços para diferentes especificações de instância, consulte **Instâncias do Cluster DCS Redis 4.0 e 5.0** e **Instâncias do Cluster Proxy DCS Redis 4.0 e 5.0** .
  > - Você pode adicionar estilhaços a uma instância de cluster referindo-se a **Modificando especificações**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

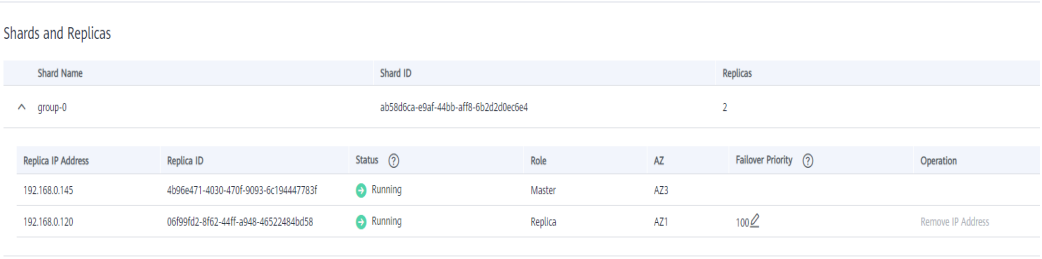
Passo 4 Clique em uma instância.

Passo 5 Clique na guia **Shards and Replicas**.

A página exibe todos os estilhaços na instância e a lista de réplicas de cada estilhaço.

Passo 6 Clique em para mostrar todas as réplicas de um estilhaço.

Figura 7-1 Listas de estilhaços e réplicas



Passo 7 Clique em **Promote to Master** na linha que contém outra réplica cuja função é **Replica**.

Passo 8 Clique em **Yes**.

----Fim

7.9 Análise de cache

7.9.1 Analisando Big Keys e Hot Keys

Ao realizar análise de chave grande e análise de chave quente, você terá uma imagem de chaves que ocupam um grande espaço e chaves que são as mais freqüentemente acessadas.

Notas sobre a análise da grande chave:

- Todas as instâncias do DCS Redis oferecem suporte à análise de grandes chaves.
- Durante a análise de grandes chaves, todas as chaves serão atravessadas. Quanto maior o número de chaves, mais tempo demora a análise.
- Realize grandes análises de chave durante as horas fora de pico e evite períodos de backup automático.
- Para uma instância principal/em espera ou cluster, a análise da chave grande é realizada no nó em espera, portanto, o impacto na instância é menor. Para uma instância de nó único, a análise de chave grande é realizada no único nó da instância e reduzirá o

desempenho de acesso à instância em até 10%. Portanto, execute grandes análises de chave em instâncias de nó único durante o horário de pico.

- Um máximo de 100 registros de análise de chave grande (20 para Strings e 80 para Lists/Sets/Zsets/Hashes) são mantidos para cada instância. Quando esse limite for atingido, os registros mais antigos serão excluídos para abrir espaço para novos registros. Você também pode excluir manualmente os registros de que não precisa mais.

Notas sobre a análise de hot key:

- Somente as instâncias do DCS Redis 4.0/5.0 oferecem suporte à análise de teclas de atalho, e o parâmetro **maxmemory-policy** das instâncias deve ser definido como **allkeys-lfu** ou **volatile-lfu**.
- Durante a análise de hot key, todas as chaves serão percorridas. Quanto maior o número de chaves, mais tempo demora a análise.
- Execute a análise de tecla quente logo após o horário de pico para garantir a precisão dos resultados da análise.
- A análise da tecla de atalho é executada no nó principal de cada instância e reduzirá o desempenho de acesso à instância em até 10%.
- Um máximo de 100 registros de análise são mantidos para cada instância. Quando esse limite for atingido, os registros mais antigos serão excluídos para abrir espaço para novos registros. Você também pode excluir manualmente os registros de que não precisa mais.

NOTA

Execute análises de teclas grandes e teclas de atalho durante horários fora de pico para evitar 100% de uso da CPU.

Procedimento para análise de Big Key

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome de uma instância do DCS Redis.

Passo 5 Escolha **Analysis and Diagnosis** > **Cache Analysis**.

Passo 6 Na página de guia **Big Key Analysis**, você pode iniciar manualmente uma análise big key ou programar uma análise automática diária.

Passo 7 Após a conclusão de uma tarefa de análise, clique em **View** para exibir os resultados da análise.

Você pode exibir os resultados da análise de diferentes tipos de dados.

NOTA

O console exibe um máximo de 20 registros de análise de chave grande para Strings e 80 para Listas, Conjuntos, Zsets e Hashes.

----Fim

Procedimento para análise de hot key

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome de uma instância do DCS Redis.

Passo 5 Escolha **Analysis and Diagnosis** > **Cache Analysis**.

Passo 6 Na página de guia **Hot Key Analysis**, você pode iniciar manualmente uma análise de hot key ou programar uma análise automática diária.

NOTA

Se a instância tiver sido criada antes de julho de 2020, o valor padrão do parâmetro **maxmemory-policy** será **noeviction**. Antes de iniciar a análise de hot key, defina este parâmetro para **allkeys-lfu** ou **volatile-lfu**. O valor padrão do parâmetro **maxmemory-policy** de uma instância criada em ou após julho de 2020 é **volatile-lru**. Para executar a análise de hot key, defina este parâmetro como **allkeys-lfu** ou **volatile-lfu** na página **Instance Configuration** > **Parameters**. Para obter detalhes sobre o **allkeys-lfu** e **volatile-lfu**, consulte **Qual é a Política de Despejo de Dados Padrão?**

Passo 7 Depois que uma tarefa de análise for concluída, clique em **View** para exibir os resultados da análise.

Os resultados da análise da tecla de atalho são exibidos.

NOTA

O console exibe um máximo de 100 registros de análise de teclas de atalho para cada instância.

Tabela 7-7 Resultados da análise de hot key

Parâmetro	Descrição
Chave	Nome de uma tecla de atalho.
Tipo	Tipo de uma tecla de atalho, que pode ser string, hash, lista, conjunto ou conjunto classificado.
Tamanho	Tamanho do valor da chave quente.
FREQ	Reflete a frequência de acesso de uma chave dentro de um período de tempo específico (geralmente 1 minuto). Para mais detalhes, visite o **site oficial do Redis**. **FREQ** é o contador de frequência de acesso logarítmico. O valor máximo de **FREQ** é 255, o que indica 1 milhão de solicitações de acesso. Depois que a **FREQ** atingir **255**, ela não será mais incrementada mesmo que as solicitações de acesso continuem aumentando. **FREQ** irá diminuir em 1 para cada minuto durante o qual a chave não é acessada.

Parâmetro	Descrição
Fragmento	Fragmento onde a tecla de atalho está localizada. **NOTA** Esse parâmetro está disponível somente para instâncias de cluster.
Banco de dados	Banco de dados onde uma tecla de atalho está localizada.

----Fim

Perguntas frequentes sobre Big Keys e Hot Keys

- **Por que a capacidade ou o desempenho de um fragmento de uma instância de cluster do Redis está sobrecarregado quando a instância ainda está abaixo do gargalo?**
- **Qual é o impacto de Big Key?**
- **Qual é o impacto de Hot Key?**
- **Como faço para evitar Big Keys e Hot Keys?**
- **Como analisar Hot Keys de uma instância do DCS Redis 3.0?**

7.9.2 Varrendo chaves expiradas

Há duas maneiras de excluir uma chave no Redis.

- Use o comando **DEL** para excluir diretamente uma chave.
- Use comandos como **EXPIRE** para definir um tempo limite em uma tecla. Após o tempo limite, a chave se torna inacessível, mas não é excluída imediatamente porque o Redis é geralmente de thread único. O Redis usa as seguintes estratégias para liberar a memória usada por chaves expiradas:
  - Eliminação livre preguiçoso: A estratégia de exclusão é controlada no loop de eventos de E/S principal. Antes de um comando de leitura/gravação ser executado, uma função é chamada para verificar se a chave a ser acessada expirou. Se tiver expirado, ele será excluído e uma resposta será retornada indicando que a chave não existe. Se a chave não tiver expirado, a execução do comando será retomada.
  - Exclusão agendada: Uma função de evento de tempo é executada em certos intervalos. Cada vez que a função é executada, uma coleção aleatória de chaves é verificada e as chaves expiradas são excluídas.

    NOTA

    Para evitar bloqueios prolongados no thread principal do Redis, nem todas as chaves são verificadas em cada evento de tempo. Em vez disso, uma coleção aleatória de chaves é verificada a cada vez. Como resultado, a memória usada por chaves expiradas não pode ser liberada rapidamente.

O DCS integra essas estratégias e permite que você libere periodicamente a memória usada por chaves expiradas. Você pode configurar varreduras programadas nos nós principais de suas instâncias. Todo o espaço de teclas é percorrido durante as verificações, acionando o Redis para verificar se as chaves expiraram e para remover chaves expiradas, se houver.

NOTA

Essa função é suportada apenas pelas instâncias do DCS Redis 4.0 e 5.0.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

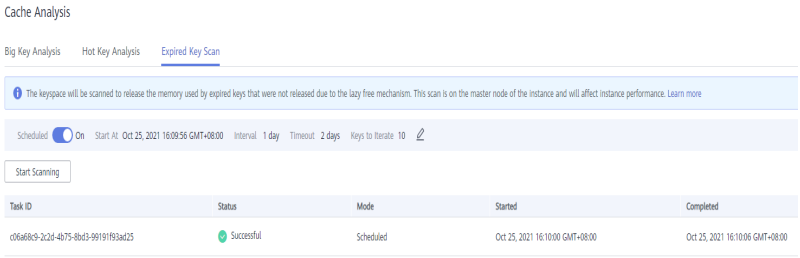
Passo 4 Clique no nome de uma instância do DCS Redis.

Passo 5 Escolha **Analysis and Diagnosis** > **Cache Analysis**.

Passo 6 Clique em **Expired Key Scan**. Você pode clicar em **Start Scanning** para verificar a instância imediatamente. Você também pode configurar uma tarefa agendada para examinar automaticamente a instância no horário especificado.

Passo 7 Depois que a tarefa de varredura de chave expirada for enviada, você poderá visualizá-la na lista de tarefas.

Figura 7-2 Tarefas de varredura de chave expiradas



----Fim

NOTA

Uma falha de varredura pode ser causada pelos seguintes problemas:

- Ocorreu uma exceção.
- A verificação expirou porque há muitas chaves. Neste caso, algumas chaves foram apagadas.

Agendamento de varreduras automáticas

Para programar varreduras automáticas, clique em  ao lado de **Scheduled**. Defina os parâmetros conforme necessário e clique em **OK**.

Tabela 7-8 descreve os parâmetros para agendamento de varreduras automáticas.

Tabela 7-8 Parâmetros para agendamento de varreduras automáticas

Parâmetro	Descrição	Intervalo de valores	Valor padrão	Observações
Começar em	A primeira varredura só pode começar após o tempo atual.	Formato: aaaa / mm / dd hh: mm: ss	-	-

Parâmetro	Descrição	Intervalo de valores	Valor padrão	Observações
Intervalo	Intervalo entre varreduras.	0 a 43.200 (unidade: minuto)	1440	● Se a varredura anterior não estiver concluída quando a hora de início chegar, a próxima varredura será ignorada. ● Se a verificação anterior for concluída dentro de cinco minutos após a hora de início, a próxima verificação não será ignorada. **NOTA** Varreduras contínuas podem causar alto uso da CPU. Defina este parâmetro com base no número total de chaves na instância e no aumento de chaves. Para obter detalhes, consulte a seguinte descrição de desempenho e sugestões de configuração.

Parâmetro	Descrição	Intervalo de valores	Valor padrão	Observações
Tempo limite	Este parâmetro é usado para impedir o tempo limite de varredura devido a razões desconhecidas. Se a verificação expirar devido a motivos desconhecidos, as tarefas agendadas subsequentes não poderão ser executadas. Após o tempo limite especificado, uma mensagem de falha é retornada e a próxima varredura será executada.	1 a 86.400 (unidade: minuto)	2880	● Defina o tempo limite para pelo menos o dobro do intervalo. ● Você pode definir um valor com base no tempo gasto em varreduras anteriores e no tempo limite máximo que pode ser tolerado no cenário do aplicativo.

Parâmetro	Descrição	Intervalo de valores	Valor padrão	Observações
Chaves para iterar	O comando **SCAN** é usado para iterar as chaves no banco de dados atual. A opção **COUNT** é usada para permitir que o usuário diga ao comando de iteração quantos elementos devem ser retornados do conjunto de dados em cada iteração. Para obter detalhes, consulte a **descrição do comando SCAN**. A varredura iterativa pode reduzir os riscos de desaceleração do Redis quando um grande número de chaves é digitalizado de cada vez.	10 a 1000	10	Por exemplo, se houver 10 milhões de chaves no Redis e o número de chaves a serem iteradas estiver definido como 1000, uma verificação completa será concluída após as iterações do 10.000.

Desempenho

- O comando **SCAN** é executado no plano de dados a cada 5 ms, ou seja, 200 vezes por segundo. Se **Keys to Iterate** estiver definida como **10**, **100**, ou **1000**, 2000, as chaves 20.000 ou 200.000 são verificadas por segundo.
- Quanto maior o número de chaves verificadas por segundo, maior o uso da CPU.

Ensaio de referência

Uma instância principal/em espera é verificada. Existem 10 milhões de chaves que não expiram e 5 milhões de chaves que expiram. O tempo de expiração é de 1 a 10 segundos.

- Eliminação natural: Os registros do 10.000 são excluídos por segundo. Demora 8 minutos para excluir 5 milhões de chaves expiradas. O uso da CPU é de cerca de 5%.

- **Keys to Iterate** definidas como **10**: A varredura leva 125 minutos (15 milhões/2000/60 segundos) e o uso da CPU é de cerca de 8%.
- **Keys to Iterate** definidas como **100**: A varredura leva 12,5 minutos (15 milhões/20.000/60 segundos) e o uso da CPU é de cerca de 20%.
- **Keys to Iterate** definidas como **1000**: A varredura leva 1,25 minutos (15 milhões/200.000/60 segundos) e o uso da CPU é de cerca de 25%.

Sugestões de configuração

- Você pode configurar o número de chaves a serem verificadas e o intervalo de varredura com base no número total de chaves e no aumento do número de chaves na instância.
- No teste de referência com 15 milhões de chaves e **Keys to Iterate** definido como **10**, a digitalização leva cerca de 125 minutos. Neste caso, defina o intervalo para mais de 4 horas.
- Se você quiser acelerar a verificação, defina **Keys to Iterate** para **100**. Demora cerca de 12,5 minutos para concluir a verificação. Portanto, defina o intervalo para mais de 30 minutos.
- Quanto maior o número de chaves para iterar, mais rápida a varredura e maior o uso da CPU. Existe um trade-off entre o tempo e o uso da CPU.
- Se o número de chaves expiradas não aumentar rapidamente, você pode verificar as chaves expiradas uma vez por dia.

NOTA

Comece a digitalizar durante as horas fora de pico. Defina o intervalo para um dia e o tempo limite para dois dias.

7.10 Exibindo consultas lentas do Redis

O Redis registra consultas que excedem um tempo de execução especificado. Você pode exibir os logs lentos no console do DCS para identificar problemas de desempenho.

Para obter detalhes sobre os comandos, visite o **site oficial do Redis**.

Configure consultas lentas com os seguintes parâmetros:

- **slowlog-log-slower-than**: O tempo máximo permitido, em microssegundos, para a execução do comando. Se esse limite for excedido, o Redis registrará o comando. O valor padrão é **10,000**. Ou seja, se a execução do comando exceder 10 ms, o comando será registrado.
- **slowlog-max-len**: O número máximo permitido de consultas lentas que podem ser registradas. O valor padrão é **128**. Ou seja, se o número de consultas lentas exceder 128, o registro mais antigo será excluído para abrir espaço para novas.

Para obter detalhes sobre os parâmetros de configuração, consulte **Modificando Parâmetros de Configuração de uma Instância**.

NOTA

Você pode exibir as consultas lentas de uma instância do DCS Redis 3.0 de cluster de proxy somente se a instância for criada após 14 de outubro de 2019.

Exibindo Consultas Lentas no Console

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome de uma instância de DCS.

Passo 5 Escolha **Analysis and Diagnosis** > **Slow Queries**.

Passo 6 Selecione uma data de início e uma data de término para exibir consultas lentas dentro do período especificado.

NOTA

- Para obter detalhes sobre os comandos, visite o **site oficial do Redis**.
- Atualmente, você pode exibir consultas lentas nos últimos sete dias.

Figura 7-3 Consultas lentas de uma instância

----Fim

7.11 Exibindo logs de execução do Redis

Você pode criar arquivos de log de execução no console do DCS para coletar logs de execução de instâncias do DCS Redis dentro de um período especificado. Depois que os logs são coletados, você pode baixar os arquivos de log para visualizar os logs.

NOTA

Essa função é suportada por instâncias do DCS Redis 4.0 e posteriores.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique em uma instância de DCS.

Passo 5 Clique na guia **Run Logs**.

Passo 6 Clique em **Create Log File** e especifique as condições de coleta.

Se a instância for do tipo principal/em espera ou cluster, você poderá especificar o estilhaço e a réplica cujos logs de execução deseja coletar. Se a instância for do tipo de nó único, os logs do único nó da instância serão coletados.

----Fim

7.12 Diagnosticando uma instância

Cenário

Se ocorrer uma falha ou um problema de desempenho, peça ao DCS para diagnosticar sua instância para saber mais sobre a causa e o impacto do problema e como lidar com ele.

Restrições

- As instâncias do DCS Redis 3.0 e do Memcached não suportam diagnóstico.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique no nome de uma instância do DCS Redis.

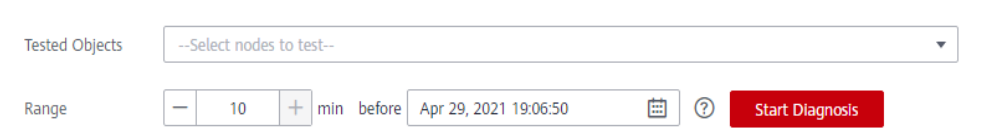
Passo 5 Escolha **Analysis and Diagnosis** > **Instance Diagnosis**.

Passo 6 Especifique o objeto e o intervalo de tempo testados e clique em **Start Diagnosis**.

- **Tested Object**: Você pode selecionar um único nó ou todos os nós.
- **Range**: Você pode especificar até 10 minutos antes de um ponto no tempo nos últimos 7 dias.

  Na figura a seguir, os dados da instância entre 18:56:50 e 19:06:50 em 7 de janeiro de 2021 serão diagnosticados.

  Figura 7-4 Especificando o objeto testado e o intervalo de tempo

Passo 7 Depois que o diagnóstico for concluído, você poderá visualizar o resultado na lista **Test History**. Se o resultado for anormal, clique em **View Report** para obter detalhes.

No relatório, você pode ver a causa e o impacto de itens anormais e sugestões para lidar com eles.

----Fim

8 Fazendo backup e restaurando instâncias

8.1 Visão geral

No console do DCS, você pode fazer backup e restaurar instâncias do DCS.

Importância do backup de instância do DCS

Há uma pequena chance de que dados sujos possam existir em uma instância do DCS devido a exceções do sistema de serviço ou problemas no carregamento de dados de arquivos de persistência. Além disso, alguns sistemas exigem não apenas alta confiabilidade, mas também segurança de dados, restauração de dados e até armazenamento permanente de dados.

Atualmente, os dados em instâncias de DCS podem ser copiados para o OBS. Se uma instância do DCS apresentar defeito, os dados da instância poderão ser restaurados do backup para que a continuidade do serviço não seja afetada.

Modos de backup

As instâncias de DCS são compatíveis com os seguintes modos de backup:

- Backup automático

  Você pode criar uma política de backup programada no console do DCS. Em seguida, os dados nas instâncias DCS escolhidas serão automaticamente copiados no horário programado.

  Você pode escolher os dias da semana em que o backup agendado será executado. Os dados de backup serão mantidos por no máximo sete dias. Os dados de backup com mais de sete dias serão automaticamente excluídos.

  O objetivo principal dos backups automatizados é criar réplicas de dados completas de instâncias de DCS para que a instância possa ser restaurada rapidamente, se necessário.

- Backup manual

  As solicitações de backup também podem ser emitidas manualmente. Em seguida, os dados nas instâncias DCS escolhidas serão permanentemente copiados para o OBS. Os dados de backup podem ser excluídos manualmente.

  Antes de executar operações de alto risco, como manutenção ou atualização do sistema, faça backup dos dados da instância do DCS.

Informações adicionais sobre backup de dados

- Tipos de instância
  - Redis: Somente instâncias principal/em espera, cluster de proxy, cluster do Redis e de divisão de leitura/gravação podem ser armazenadas em backup e restauradas, enquanto instâncias de nó único não podem. No entanto, você pode exportar dados de uma instância de nó único para um arquivo RDB usando redis-cli. Para obter detalhes, consulte **Como exportar dados de instância do DCS Redis?**
  - Memcached: Somente as instâncias principal/em espera podem ser armazenadas em backup e restauradas, enquanto as instâncias de nó único não podem.
- Mecanismos de backup

  O DCS for Redis 3.0 mantém os dados em arquivos AOF. Os DCS para Redis 4.0 e 5.0 mantêm os dados em arquivos RDB ou AOF no modo de backup manual e em arquivos RDB no modo de backup automático.

  Para exportar arquivos de backup RDB de instâncias do DCS Redis 3.0, execute o comando **redis-cli -h** *{redis_address}* **-p 6379 [-a** *{password}***] --rdb** *{output.rdb}* no redis-cli.

  NOTA

  - O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.
  - Para uma instância do DCS Redis 3.0 de nó único na qual o comando **SYNC** pode ser executado, você pode executar esse comando para exportar o arquivo RDB. Para uma instância do Cluster de Proxy DCS Redis 3.0, o comando **SYNC** não pode ser executado devido à arquitetura. Portanto, o arquivo RDB não pode ser exportado.

  As tarefas de backup são executadas em nós de cache em espera. O backup dos dados da instância do DCS é feito ao compactar e armazenar os arquivos de persistência de dados do nó de cache em espera para o OBS.

  O DCS verifica as políticas de backup da instância uma vez por hora. Se uma política de backup for correspondida, o DCS executará uma tarefa de backup para a instância do DCS correspondente.
- Impacto nas instâncias de DCS durante o backup

  As tarefas de backup são executadas em nós de cache em espera, sem incorrer em tempo de inatividade.

  No caso de sincronização completa de dados ou carga pesada de instância, leva alguns minutos para concluir a sincronização de dados. Se o backup da instância for iniciado antes da conclusão da sincronização de dados, os dados de backup ficarão um pouco atrás dos dados no nó do cache principal.

  Durante o backup da instância, o nó de cache em espera deixa de persistir as alterações mais recentes nos arquivos de disco. Se novos dados forem gravados no nó de cache principal durante o backup, o arquivo de backup não conterá os novos dados.
- Tempo de backup

  É aconselhável fazer backup dos dados da instância durante períodos fora de pico.
- Armazenamento e preço dos arquivos de backup

  Os arquivos de backup são armazenados no OBS.

  A DCS fornece o serviço de backup gratuito, mas as cobranças do OBS serão incorridas pela quantidade e pelo período em que o espaço de armazenamento é consumido.
- Tratamento de exceções no backup programado

Se uma tarefa de backup agendada for acionada enquanto a instância do DCS estiver reiniciando ou sendo ampliada, a tarefa de backup agendada será executada no próximo ciclo.

Se o backup de uma instância do DCS falhar ou o backup for adiado porque outra tarefa está em andamento, o DCS tentará fazer o backup da instância no próximo ciclo. Um máximo de três tentativas são permitidas dentro de um único dia.

- Período de retenção de dados de backup

  Os arquivos de backup agendados são retidos por até sete dias. Você pode configurar o período de retenção. No final do período de retenção, a maioria dos arquivos de backup da instância do DCS será excluída automaticamente, mas pelo menos um arquivo de backup será mantido.

  Os arquivos de backup manuais são mantidos permanentemente e precisam ser excluídos manualmente.

restauração de dados

- Processo de restauração de dados

  a. Você pode iniciar uma solicitação de restauração de dados usando o console do DCS.
  b. O DCS obtém o arquivo de backup do OBS.
  c. A leitura/gravação na instância do DCS é suspensa.
  d. O arquivo de persistência de dados original do nó de cache principal é substituído pelo arquivo de backup.
  e. O novo arquivo de persistência de dados (ou seja, o arquivo de backup) é recarregado.
  f. Os dados são restaurados e a instância do DCS começa a fornecer o serviço de leitura/gravação novamente.

- Impacto nos sistemas de serviço

  As tarefas de restauração são executadas em nós de cache principal. Durante a restauração, os dados não podem ser gravados ou lidos de instâncias.

- Tratamento de exceções de restauração de dados

  Se um arquivo de backup estiver corrompido, o DCS tentará corrigir o arquivo de backup ao restaurar os dados da instância. Se o arquivo de backup for corrigido com êxito, a restauração prosseguirá. Se o arquivo de backup não puder ser corrigido, a instância do DCS principal/em espera será alterada de volta para o estado em que estava antes da restauração de dados.

8.2 Configurando uma política de backup

No console DCS, você pode configurar uma política de backup automático. Em seguida, o sistema faz backup dos dados em suas instâncias de acordo com a política de backup.

Se o backup automático não for necessário, desative a função de backup automático na política de backup.

Pré-requisitos

Uma instância de DCS de divisão principal/em espera, cluster ou leitura/gravação está no estado **Running**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Filtre instâncias de DCS para localizar a instância de DCS desejada. Atualmente, você pode pesquisar instâncias por nome, especificação, ID, endereço IP, AZ, status, tipo de instância, mecanismo de cache e muitos outros atributos.

Passo 4 Clique no nome da instância do DCS desejada para ir para a página de detalhes.

Passo 5 Na página de detalhes da instância, clique em **Backups & Restorations**.

Passo 6 Deslize para a direita para ativar o backup automático. As políticas de backup serão exibidas.

Tabela 8-1 Parâmetros em uma política de backup

Parâmetro	Descrição
Agendamento de backup	Dia da semana em que os dados da instância do DCS escolhida são automaticamente copiados. Você pode selecionar um ou vários dias da semana.
Período de retenção (dias)	O número de dias em que o backup automático dos dados é mantido. Os dados de backup serão excluídos permanentemente no final do período de retenção e não poderão ser restaurados. Faixa de valor: 1–7.
Hora de início	Hora em que o backup automático é iniciado. Valor: a hora cheia entre 00:00 e 23:00 O DCS verifica as políticas de backup uma vez a cada hora. Se a hora de início do backup em uma política de backup tiver chegado, os dados na instância correspondente serão copiados. **NOTA** O backup da instância leva de 5 a 30 minutos. Os dados adicionados ou modificados durante o processo de backup não serão copiados. Para reduzir o impacto do backup nos serviços, recomenda-se que o backup dos dados seja feito durante os períodos fora de pico. Somente instâncias no estado **Running** podem ser armazenadas em backup.

Passo 7 Clique em **OK**.

----Fim

8.3 Fazendo backup manual de uma instância de DCS

Você pode fazer backup manual de dados em instâncias de DCS em tempo hábil. Esta seção descreve como fazer backup manual de dados em instâncias principal/em espera usando o console do DCS.

Por padrão, o backup manual dos dados é mantido permanentemente. Se os dados de backup não estiverem mais em uso, você poderá excluí-los manualmente.

Pré-requisitos

Uma instância de DCS de divisão principal/em espera, cluster ou leitura/gravação está no estado **Running**.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Filtre instâncias de DCS para localizar a instância de DCS desejada. Atualmente, você pode pesquisar instâncias por nome, especificação, ID, endereço IP, AZ, status, tipo de instância, mecanismo de cache e muitos outros atributos.

Passo 4 Clique no nome da instância do DCS desejada para ir para a página de detalhes.

Passo 5 Na página de detalhes da instância, clique em **Backups & Restorations**.

Passo 6 Clique em **Create Backup**.

Passo 7 Selecione um formato de arquivo de backup.

Somente as instâncias do DCS Redis 4.0/5.0 oferecem suporte à seleção de formato de arquivo de backup.

Passo 8 Na caixa de diálogo **Create Backup**, clique em **OK**.

As informações na caixa de texto **Description** não podem exceder 128 bytes.

NOTA

O backup da instância leva de 10 a 15 minutos. Os dados adicionados ou modificados durante o processo de backup não serão copiados.

----Fim

8.4 Restaurando uma instância de DCS

No console do DCS, você pode restaurar dados de backup para uma instância do DCS escolhida.

Pré-requisitos

- Uma instância de DCS de divisão principal/em espera, cluster ou leitura/gravação está no estado **Running**.
- Uma tarefa de backup foi executada para fazer backup dos dados na instância a ser restaurada e a tarefa de backup foi bem-sucedida.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Filtre instâncias de DCS para localizar a instância de DCS desejada. Atualmente, você pode pesquisar instâncias por nome, especificação, ID, endereço IP, AZ, status, tipo de instância, mecanismo de cache e muitos outros atributos.

Passo 4 Clique no nome da instância do DCS desejada para ir para a página de detalhes.

Passo 5 Na página de detalhes da instância, clique em **Backups & Restorations**.

Em seguida, é exibida uma lista de tarefas de backup histórico.

Passo 6 Clique em **Restore** na linha que contém a tarefa de backup escolhida.

Passo 7 Clique em **OK** para iniciar a restauração da instância.

As informações na caixa de texto **Description** não podem exceder 128 bytes.

Você pode exibir os resultados de todas as tarefas de restauração na página **Restoration History**. Os registros não podem ser apagados.

NOTA

A restauração da instância leva de 1 a 30 minutos.

Enquanto estão sendo restauradas, as instâncias de DCS não aceitam solicitações de operação de dados de clientes porque os dados existentes estão sendo substituídos pelos dados de backup.

----Fim

8.5 Baixando um arquivo de backup RDB ou AOF

Os dados de backup automático podem ser retidos por no máximo 7 dias. O backup manual de dados não é gratuito e ocupa espaço no OBS. Devido a essas limitações, é aconselhável baixar os arquivos de backup RDB e AOF e salvá-los permanentemente no host local.

Essa função é suportada apenas por instâncias principal/em espera, e não por instâncias de nó único. Para exportar os dados de uma instância de nó único para um arquivo RDB, você pode usar o redis-cli. Para obter detalhes, consulte **Como exportar dados de instância do DCS Redis?**

Para exportar os dados de uma instância principal/em espera ou de cluster, faça o seguinte:

- Redis 3.0 Exporte os dados da instância para arquivos AOF usando o console DCS ou para arquivos RDB executando o **redis-cli -h** *{redis_address}* **-p 6379 [-a** *{password}***] --rdb** *{output.rdb}* usando redis-cli.

  NOTA

  O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS para Redis 4.0 ou 5.0.

- Redis 4.0 e 5.0: Exporte os dados da instância para arquivos AOF ou RDB usando o console DCS.

Pré-requisitos

O backup da instância foi feito e o backup ainda é válido.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Filtre instâncias de DCS para localizar a instância de DCS desejada. Atualmente, você pode pesquisar por nome, especificação, ID, endereço IP, AZ, status, tipo de instância, mecanismo de cache, projeto empresarial, modo de faturamento e tags.

Passo 4 Clique no nome da instância do DCS para exibir mais detalhes sobre a instância do DCS.

Passo 5 Na página de detalhes da instância, clique em **Backups & Restorations**.

Em seguida, é exibida uma lista de tarefas de backup histórico.

Passo 6 Clique em **Download** na linha que contém a tarefa de backup escolhida.

Passo 7 Na caixa de diálogo **Download Backup File** exibida, selecione um dos dois métodos de download a seguir.

Métodos de download:

- Por URL
  a. Passo 1: Defina o período de validade da URL e clique em **Query**.
  b. Passo 2: Baixe o arquivo de backup usando a lista das URL.

  NOTA

  Se você optar por copiar os URL, use aspas para citar os URL ao executar o comando **wget** no Linux. Por exemplo:

  wget 'https://obsEndpoint.com:443/redisdemo.rdb?parm01=value01&parm02=value02'

  Isso ocorre porque a URL contém o caractere especial e (&), que irá confundir o comando **wget**. Citar a URL facilita a identificação da URL.

- Por OBS

Siga o procedimento exibido.

----Fim

9 Migrando dados da instância

9.1 Visão geral da migração de dados

O console DCS suporta migração on-line (total ou incrementalmente) e migração de backup (importando arquivos de backup) com operações intuitivas.

- A migração de backup é adequada quando as instâncias do Redis de origem e de destino não estão conectadas e a instância do Redis de origem não suporta os comandos **SYNC** e **PSYNC**. Para migrar dados, importe seus arquivos de backup para o OBS, e o DCS lerá os dados do OBS e migrará os dados para a instância do DCS Redis de destino. Como alternativa, você pode importar os arquivos de backup diretamente para a instância do DCS.
- A migração on-line é adequada quando a instância do Redis de origem suporta os comandos **SYNC** e **PSYNC**. Os dados na instância do Redis de origem podem ser migrados de forma integral ou incremental para a instância de destino.

  Durante a migração on-line, o comando **PSYNC** é entregue ao endereço de origem. Para obter detalhes sobre como isso funciona, consulte a **explicação de replicação**. Este comando causará uma operação de fork na extremidade de origem, o que afeta a latência. Para obter detalhes sobre o escopo do impacto, consulte o **site oficial do Redis**.

NOTA

Atualmente, a função de migração de dados é gratuita no OBT. Você será notificado quando a migração de dados começar a ser cobrada.

Para obter mais informações sobre ferramentas e esquemas de migração, consulte **Ferramentas e esquemas de migração**

Tabela 9-1 Modos de migração de dados DCS

Modo de migração	Origem	Alvo: DCS		
		Nó único e principal/em espera	Cluster de proxy	Cluster do Redis

Importando arquivos de backup	Bucket de OBS: arquivos AOF **NOTA** Arquivos AOF exportados de instâncias Redis 4.0/5.0 da HUAWEI CLOUD e de outras instâncias com compactação RDB ativada não podem ser importados.	√	√	×
	Bucket de OBS: arquivos RDB	√	√	√
Migração de dados online	DCS para Redis: nó único ou principal/em espera	√	√	√
	DCS for Redis: Cluster de proxy **NOTA** As instâncias do Cluster de Proxy DCS Redis 3.0 não podem ser usadas como origem, enquanto as instâncias do Cluster de Proxy DCS Redis 4.0 ou 5.0 podem.	√	√	√
	DCS for Redis: Cluster do Redis	√	√	√
	Redis de nó único auto-hospedado ou principal/em espera	√	√	√
	Cluster baseado em proxy auto-hospedado Redis	√	√	√
	Cluster do Redis auto-hospedado	√	√	√
	Outros Redis: nó único ou principal/em espera	×	×	×
	Outros Redis: cluster baseado em proxy	×	×	×
	Outros Redis: Cluster do Redis	×	×	×

NOTA

- **DCS for Redis** refere-se a instâncias do Redis fornecidas pelo DCS
- **Self-hosted Redis** refere-se ao Redis auto-hospedado na nuvem, de outros fornecedores de nuvem ou em centros de dados locais.
- **Other cloud Redis** refere-se aos serviços Redis fornecidos por outros fornecedores de nuvem.
- √: Compatível. ×: Incompatível.
- Você pode migrar dados on-line de forma completa ou incremental de **other cloud Redis** para **DCS for Redis** se eles estiverem conectados e os comandos **SYNC** e **PSYNC** puderem ser executados no Redis de origem. No entanto, algumas instâncias fornecidas por outros fornecedores de nuvem podem não ser migradas on-line. Nesse caso, migre dados por meio da importação de backup ou use outros esquemas de migração. Para obter detalhes, consulte **Ferramentas e esquemas de migração** .

9.2 Importando arquivos de backup de um bucket do OBS

Cenário

Use o console do DCS para migrar dados do Redis do Redis de outra nuvem ou do Redis auto-hospedado para o HUAWEI CLOUD DCS for Redis.

Basta fazer o download dos dados de origem do Redis e, em seguida, fazer o upload dos dados para um bucket do OBS na mesma região da instância do DCS Redis de destino. Depois de criar uma tarefa de migração no console do DCS, o DCS lerá os dados do bucket do OBS e os dados serão migrados para a instância de destino.

Os arquivos .aof, .rbb, .zip e .tar.gz podem ser carregados em intervalos do OBS. Você pode fazer upload diretamente dos arquivos .aof e .rdb ou compactá-los em arquivos .zip ou .tar.gz antes de fazer o upload.

Pré-requisitos

- O bucket do OBS deve estar na mesma região que a instância do DCS Redis de destino.
- Os arquivos de dados a serem carregados devem estar no formato .aof, .rdb, .zip ou .tar.gz.
- Para migrar dados de uma instância do Redis de nó único ou principal/em espera de outra nuvem, crie uma tarefa de backup e baixe o arquivo de backup.
- Para migrar dados de uma instância de cluster do Redis de outra nuvem, faça download de todos os arquivos de backup, carregue todos eles no intervalo do OBS e selecione todos eles para a migração. Cada arquivo de backup contém dados para um fragmento da instância.
- Os arquivos de backup .rdb do Redis 5.0 auto-hospedado não podem ser importados. Os arquivos de backup .rdb do Redis 3.0 ou 4.0 auto-hospedado podem ser exportados usando o redis-cli. Os arquivos .rdb de outros Redis na nuvem podem ser exportados apenas criando tarefas de backup e não podem ser exportados executando comandos no redis-cli.
- Instâncias de cluster do Redis suportam apenas arquivos .rdb.

Passo 1: Preparar a instância do DCS Redis de destino

- Se uma instância do DCS Redis de destino não estiver disponível, crie uma primeiro. Para obter detalhes, consulte **Comprando uma instância do DCS Redis**.
- Se você já tiver uma instância do DCS Redis, não precisará criar uma novamente, mas precisará limpar os dados da instância antes da migração. Para obter detalhes, consulte **Limpando dados de instância do DCS**.

Você pode usar uma instância do DCS Redis 3.0, 4.0 ou 5.0 como a instância de destino.

Passo 2: Criar um bucket do OBS e fazer upload de arquivos de backup

Passo 1 Carregue os arquivos de dados de backup para o bucket do OBS usando o OBS Browser+.

Se o arquivo de backup a ser carregado for menor que 5 GB, vá para a etapa **Passo 2** para carregar o arquivo usando o console do OBS.

Se o arquivo de backup a ser carregado for maior que 5 GB, siga as **instruções** fornecidas pelo OBS.

Passo 2 No console do OBS, carregue os arquivos de dados de backup para o bucket do OBS.

Execute as seguintes etapas se os arquivos de backup forem menores que 5 GB:

1. Crie um bucket do OBS.

   Ao criar um bucket do OBS, preste atenção à configuração dos seguintes parâmetros. Para obter detalhes sobre como definir outros parâmetros, consulte **Criando um Bucket** no *Guia do Usuário de OBS*.

   a. **Region**:

      O bucket do OBS deve estar na mesma região que a instância do DCS Redis de destino.

   b. **Storage Class**: Selecione Acesso **Standard** ou **Infrequent Access**.

      Não selecione **Archive**. Caso contrário, a migração falhará.
2. Na lista de intervalos, clique no intervalo criado em **Passo 2.1**.
3. No painel de navegação, escolha **Objects**.
4. Na página de guia **Objects**, clique em **Upload Object**.
5. Especifique **Storage Class**.

   Não selecione **Archive**. Caso contrário, a migração falhará.
6. Carregar os objetos.

   Arraste arquivos ou pastas para a área **Upload Object** ou clique em **add file**.

   Um máximo de 100 arquivos podem ser carregados por vez. O tamanho total não pode exceder 5 GB.

Figura 9-1 Carregando objetos em lotes

7. (Opcional) Selecione **KMS encryption** para criptografar os arquivos carregados.
8. Clique em **Upload**.

----Fim

Passo 3: Criar uma Tarefa de Migração

Passo 1 Efetue login no console do DCS.

Passo 2 No painel de navegação, escolha **Data Migration**.

Passo 3 Clique em **Create Backup Import Task**.

Passo 4 Informe o nome e a descrição da tarefa.

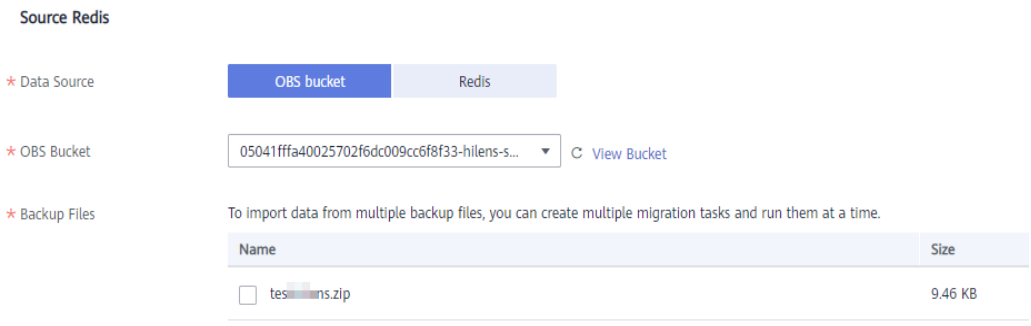
Passo 5 Na área **Source Redis**, selecione **OBS Bucket** para **Data Source** e, em seguida, selecione o bucket do OBS para o qual você fez upload dos arquivos de backup.

Na tabela **Backup Files**, os arquivos que você carregou são exibidos.

NOTA

Você pode carregar arquivos no formato .aof, .rdb, .zip ou .tar.gz.

Figura 9-2 Especificando as informações do arquivo de backup



Passo 6 Selecione os arquivos de backup cujos dados serão migrados.

Passo 7 Selecione a instância do Redis de destino preparada em **Passo 1: Preparar a instância do DCS Redis de destino**. Se a instância do Redis de destino tiver uma senha, digite a senha e teste a conexão para verificar se a senha está correta.

Passo 8 Clique em **Next**.

Passo 9 Confirme os detalhes da tarefa de migração e clique em **Submit**.

Volte para a lista de tarefas de migração de dados. Depois que a migração for bem-sucedida, o status da tarefa será alterado para **Successful**.

----Fim

9.3 Importando arquivos de backup do Redis

Cenário

Você pode migrar dados de backup do Redis para instâncias de DCS Redis principal/em espera ou de cluster.

Basta fazer backup dos dados do Redis, criar uma tarefa de migração no console do DCS e importar o backup para uma instância do DCS Redis.

Pré-requisitos

Uma instância do DCS Redis de destino foi criada como o destino para a migração. A instância de origem tem dados gravados e foi feita uma cópia de segurança.

Passo 1: Obter o Nome da Instância de Origem

Obtenha o nome da instância do Redis de origem.

Passo 2: Preparar a instância do DCS Redis de destino

- Se uma instância do DCS Redis não estiver disponível, crie uma primeiro. Para mais detalhes, consulte **Comprando uma Instância DCS Redis**.
- Se uma instância do DCS Redis estiver disponível, não será necessário criar uma nova. No entanto, você deve limpar os dados da instância antes da migração. Para mais detalhes, consulte **Limpando dados de instância do DCS**.

Você pode usar uma instância do DCS Redis 3.0, 4.0 ou 5.0 como a instância de destino.

Passo 3: Criar uma Tarefa de Migração

Passo 1 Efetue login no console do DCS.

Passo 2 No painel de navegação, escolha **Data Migration**. A lista de tarefas de migração é exibida.

Passo 3 Clique em **Create Backup Import Task**.

Passo 4 Informe o nome e a descrição da tarefa.

Passo 5 Defina **Data Source** como **Redis**.

Passo 6 Para **Source Redis Instance**, selecione a instância preparada em **Passo 1: Obter o Nome da Instância de Origem**.

Passo 7 Selecione a tarefa de backup cujos dados serão migrados.

Passo 8 Selecione a instância de destino criada em **Passo 2: Preparar a instância do DCS Redis de destino**.

Passo 9 Informe a senha da instância de destino. Clique em **Test Connection** para verificar a senha.

Passo 10 Clique em **Next**.

Passo 11 Confirme os detalhes da tarefa de migração e clique em **Submit**.

Volte para a lista de tarefas de migração de dados. Depois que a migração for bem-sucedida, o status da tarefa será alterado para **Successful**.

----Fim

9.4 Migração online

Cenário

Se as instâncias de origem e de destino estiverem interconectadas e os comandos **SYNC** e **PSYNC** forem suportados pela instância de origem, os dados poderão ser migrados online de forma integral ou incremental da origem para o destino.

⚠ CUIDADO

- Se os comandos **SYNC** e **PSYNC** estiverem desativados na instância do Redis de origem, ative-os antes de executar a migração online. Caso contrário, a migração falhará. Se você usar uma instância do HUAWEI CLOUD DCS Redis para migração online, o comando **SYNC** será ativado automaticamente.
- Você não pode usar redes públicas para migração online.
- Durante a migração online, é recomendável definir **repl-timeout** na instância de origem para 300s e **client-output-buffer-limit** para 20% da memória máxima da instância.

NOTA

Durante a migração online, os resultados dos comandos **FLUSHDB** e **FLUSHALL** executados na origem não serão sincronizados com o destino.

Impactos nos serviços

Durante a migração online, os dados são essencialmente sincronizados na íntegra para uma nova réplica. Portanto, realize a migração online durante as horas de baixa demanda.

Pré-requisitos

- Antes de migrar dados, leia **Ferramentas e esquemas de migração** para saber mais sobre a função de migração de dados DCS e selecione uma instância de destino apropriada.

- Por predefinição, uma instância de Cluster de Proxy tem apenas um banco de dados (DB0). Antes de migrar dados de uma instância de nó único ou principal/em espera para uma instância de Cluster de Proxy, verifique se existem dados em bancos de dados diferentes do DB0. Se sim, habilite multi-DB para a instância de Cluster de Proxy consultando **Ativando o Multi-DB**.
- Por padrão, uma instância de Cluster do Redis tem apenas um DB (DB0). Antes de migrar dados de uma instância de nó único ou principal/em espera para uma instância do Cluster do Redis, verifique se existem dados em bancos de dados diferentes do DB0. Para garantir que a migração seja bem-sucedida, mova todos os dados para o DB0 consultando **Migração online com Rump**.

Passo 1: Obter informações sobre a instância do Redis de origem

- Se a origem for uma instância do Cloud Redis, obtenha seu nome.
- Se a origem for um Redis auto-hospedado, obtenha seu endereço IP ou nome de domínio e número de porta.

Passo 2: Preparar a instância do DCS Redis de destino

- Se uma instância do DCS Redis de destino não estiver disponível, crie uma primeiro. Para obter detalhes, consulte **Comprando uma instância do DCS Redis**.
- Se você já tiver uma instância do DCS Redis, não precisará criar uma novamente, mas precisará limpar os dados da instância antes da migração. Para obter detalhes, consulte **Limpando dados de instância do DCS**.

  Se os dados da instância de destino não forem apagados antes da migração e as instâncias de origem e de destino contiverem a mesma chave, a chave na instância de destino será substituída pela chave na instância de origem após a migração.

Passo 3: Verifique a rede

Passo 1 Verifique se a instância do Redis de origem, a instância do Redis de destino e a tarefa de migração estão configuradas com a mesma VPC.

Se sim, vá para **Passo 4: Criar uma Tarefa de Migração Online**. Se não, vá para **Passo 2**.

Passo 2 Verifique se as VPC configuradas para a instância do Redis de origem, a instância do Redis de destino e a tarefa de migração estão conectadas para garantir que o recurso de VM da tarefa de migração possa acessar as instâncias do Redis de origem e de destino.

Se sim, vá para **Passo 4: Criar uma Tarefa de Migração Online**. Se não, vá para **Passo 3**.

Passo 3 Execute as seguintes operações para estabelecer a rede.

- Se as instâncias do Redis de origem e de destino estiverem na mesma região, crie uma conexão de emparelhamento de VPC referindo-se a **Conexão emparelhamento VPC**.
- Se as instâncias Redis de origem e de destino estiverem em regiões diferentes, crie uma conexão de nuvem consultando **Cloud Connect Primeiros passos**.
- Se as instâncias do Redis de origem e de destino estiverem em nuvens diferentes, crie uma conexão consultando **Documentação do Direct Connect**.

----Fim

Passo 4: Criar uma Tarefa de Migração Online

Passo 1 Efetue login no console do DCS.

Passo 2 No painel de navegação, escolha **Data Migration**.

Passo 3 Clique em **Create Online Migration Task**.

Passo 4 Informe o nome e a descrição da tarefa.

Passo 5 Configure a VPC, a sub-rede e o grupo de segurança para a tarefa de migração.

A VPC, a sub-rede e o grupo de segurança facilitam a migração. Certifique-se de que os recursos de migração possam acessar as instâncias do Redis de origem e de destino.

----Fim

Passo 5: Configurar a Tarefa de Migração Online

Passo 1 Na página de guia **Online Migration**, clique em **Configure** na linha que contém a tarefa de migração online que você acabou de criar.

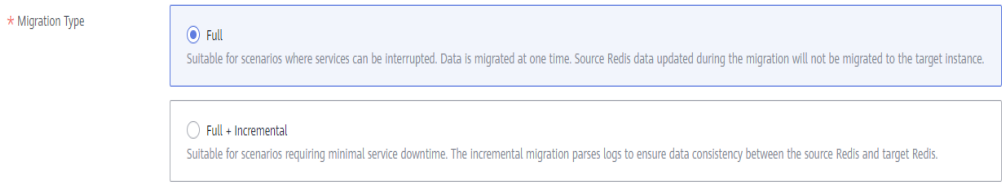
Passo 2 Selecione um tipo de migração.

Os tipos de migração suportados são **Full** e **Full + Incremental**, que são descritos em **Tabela 9-2**.

Tabela 9-2 Descrição do tipo de migração

Tipo de migração	Descrição
Completo	Adequado para cenários em que os serviços podem ser interrompidos. Os dados são migrados de uma só vez. **Os dados da instância de origem atualizados durante a migração não serão migrados para a instância de destino.**
Total incremento incremental	Adequado para cenários que exigem tempo mínimo de inatividade do serviço. A migração incremental analisa os logs para garantir a consistência dos dados entre as instâncias de origem e de destino. Quando a migração for iniciada, ela permanecerá **Migrating** até que você clique em **Stop** na coluna **Operation**. Depois que a migração for interrompida, os dados na instância de origem não serão perdidos, mas os dados não serão gravados na instância de destino. Quando a rede de transmissão está estável, o atraso da migração incremental ocorre em segundos. O atraso real depende da qualidade de transmissão do link de rede.

Figura 9-3 Selecionando o tipo de migração



Passo 3 Configurar o Redis de origem e o Redis de destino.

1. **Source Redis Type**: Selecione **Redis in the cloud** ou **Self-hosted Redis**, conforme necessário.
   - **Redis in the cloud**: uma instância do Redis DCS da HUAWEI CLOUD que está na mesma VPC que a tarefa de migração
   - **Self-hosted Redis**: Redis auto-hospedado na HUAWEI CLOUD, em outra nuvem ou em data centers locais. Se você selecionar essa opção, insira os endereços do Redis.
2. Se a instância estiver protegida por senha, você pode clicar em **Test Connection** para verificar se a senha da instância está correta e se a rede está conectada.

Passo 4 Para **Target Redis Instance**, selecione a Instância do Redis do DCS preparada em **Passo 2: Preparar a instância do DCS Redis de destino**.

Se a instância estiver protegida por senha, você pode clicar em **Test Connection** para verificar se a senha da instância atende aos requisitos.

NOTA

Se as instâncias Redis de origem e de destino estiverem conectadas, mas estiverem em regiões diferentes da HUAWEI CLOUD, você só poderá selecionar **Self-hosted Redis** for **Target Redis Type** e inserir os endereços das instâncias, independentemente de a instância do Redis de destino ser auto-hospedada ou na nuvem.

Passo 5 Confirme os detalhes da tarefa de migração e clique em **Submit**.

Volte para a lista de tarefas de migração de dados. Depois que a migração for bem-sucedida, o status da tarefa será alterado para **Successful**.

NOTA

Quando a migração incremental for iniciada, ela permanecerá em **Migrating** até que você clique em **Stop**.

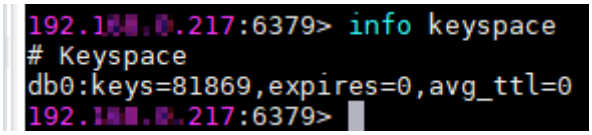
Se a migração falhar, clique na tarefa de migração e verifique o log na página **Migration Logs**.

----Fim

Verificando a migração

Após a conclusão da migração, use o redis-cli para conectar as instâncias do Redis de origem e de destino para verificar a integridade dos dados.

1. Conecte-se ao Redis de origem e ao Redis de destino.
2. Execute o comando **info keyspace** para verificar os valores das **keys** e **expires**.



3. Calcule a diferença entre os valores de **keys** e **expires** do Redis de origem e do Redis de destino. Se as diferenças forem as mesmas, os dados estão completos e a migração é bem-sucedida.

Durante a migração completa, os dados do Redis de origem atualizados durante a migração não serão migrados para a instância de destino.

9.5 Comutação IP

Cenário

No momento, você não pode alterar o tipo de instância ao usar a função de modificação de especificação. Para modificar as especificações da instância ao alterar o tipo de instância, você pode executar a alternância de IP após a migração de dados. Ao alternar endereços IP, você também pode alterar a arquitetura AZ e CPU usada por uma instância.

- Depois que a migração de dados online for concluída, você poderá alternar os endereços IP.
- Os endereços de IP podem ser revertidos conforme necessário após a comutação.

NOTA

- A comutação IP é suportada apenas pelas instâncias do DCS Redis 4.0 e 5.0.
- A comutação de IP é suportada apenas quando as instâncias de origem e de destino são instâncias do Redis na nuvem.

Pré-requisitos

- Obtenha informações sobre as instâncias de origem e de destino. Para obter detalhes sobre a preparação de uma instância de destino, consulte **Passo 2: Preparar a instância do DCS Redis de destino**.
- Certifique-se de que as instâncias de origem e de destino possam se comunicar umas com as outras. Para mais detalhes, consulte **Passo 3: Verifique a rede**.
- As instâncias de destino e de origem devem usar a mesma porta.
- A comutação IP só pode ser realizada quando as seguintes condições forem atendidas:
  - A comutação IP depende da função de migração de dados. Portanto, as instâncias de origem e de destino devem suportar a função de migração de dados. Para mais detalhes, consulte **Tabela 9-1**.
  - Tanto as instâncias de origem quanto as de destino são instâncias do Redis na nuvem.
  - **Tabela 9-3** lista os cenários de comutação IP suportados.

Tabela 9-3 Cenários de comutação IP

Origem	Alvo
Divisão de nó único, principal/em espera ou leitura/gravação	Nó único, principal/em espera, divisão de leitura/gravação ou cluster de proxy
Cluster de proxy	Nó único, principal/em espera, divisão de leitura/gravação ou cluster de proxy

Precauções para IP Switching

1. A migração online será interrompida durante a comutação.
2. As instâncias serão somente leitura por um minuto e desconectadas por vários segundos durante a comutação.
3. Se o aplicativo não puder se reconectar ao Redis ou lidar com exceções, talvez seja necessário reiniciar o aplicativo após a comutação de IP.
4. Se as instâncias de origem e de destino estiverem em sub-redes diferentes, as informações de sub-rede serão atualizadas após a comutação.
5. Se a origem for uma instância principal/em espera, o endereço IP do nó em espera não será comutado. Certifique-se de que esse endereço IP não seja usado por seus aplicativos.
6. Se seus aplicativos usarem um nome de domínio para se conectar ao Redis, o nome de domínio será usado para a instância de origem. Selecione **Yes** para **Switch Domain Name**.
7. Certifique-se de que as senhas das instâncias de origem e de destino sejam as mesmas. Se forem diferentes, a verificação falhará após a troca.
8. Se uma lista de permissões estiver configurada para a instância de origem, certifique-se de que a mesma lista de permissões esteja configurada para a instância de destino antes de alternar os endereços IP.

Alternando endereços IP

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Data Migration**.

Passo 4 Clique em **Create Online Migration Task**.

Passo 5 Informe o nome e a descrição da tarefa.

Passo 6 Configure a VPC, a sub-rede e o grupo de segurança para a tarefa de migração.

A VPC, a sub-rede e o grupo de segurança facilitam a migração. Certifique-se de que os recursos de migração possam acessar as instâncias do Redis de origem e de destino.

Passo 7 Configure a tarefa de migração consultando **Configurar a Tarefa de Migração Online**. Defina **Migration Type** como **Full + Incremental**.

Passo 8 Na página **Online Migration**, quando o status da tarefa de migração for alterado para **Incremental migration in progress**, escolha **More** > **Switch IP** na coluna **Operation**.

Passo 9 Na caixa de diálogo **Switch IP**, selecione se deseja alternar o nome de domínio.

NOTA

- Se um nome de domínio for usado, comute-o. Caso contrário, você deverá modificá-lo no cliente.
- Se nenhum nome de domínio for usado, o DNS das instâncias será atualizado.

Passo 10 Clique em **OK**. A tarefa de comutação de endereços IP foi enviada com êxito. Quando o status da tarefa de migração for alterado para **IP switched**, a troca de endereço IP será concluída.

----Fim

Rolling Back Endereços IP

Se você quiser alterar o endereço IP da instância para o endereço IP original, execute as seguintes operações:

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Data Migration**.

Passo 4 Na página **Online Migration**, localize a linha que contém a tarefa de migração no estado **IP switched**, escolha **More** > **Roll Back IP**.

Passo 5 Na caixa de diálogo de confirmação, clique em **Yes**. A tarefa de reversão do endereço IP foi enviada com sucesso. Quando o status da tarefa muda para **IP rolled back**, a reversão é concluída.

----Fim

10 Modelos de parâmetros

10.1 Exibindo Modelos de Parâmetros

Esta seção descreve como exibir modelos de parâmetros no console do DCS.

Procedimento

Passo 1 Efetue login no console do DCS.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Parameter Templates**.

Passo 4 Escolha a guia **Default Templates** ou **Custom Templates**.

Passo 5 Exibir modelos de parâmetros.

Atualmente, você pode inserir uma palavra-chave na caixa de pesquisa para pesquisar um modelo de parâmetro pelo nome do modelo.

Passo 6 Clique em um modelo de parâmetro. Os parâmetros contidos no modelo são exibidos. Para obter detalhes sobre os parâmetros, consulte **Tabela 10-1**.

Tabela 10-1 Parâmetros de configuração da instância do DCS Redis

Parameter	Descrição	Value Range	Default Value
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0
apêndicefsync	Controla a frequência com que o fsync() transfere dados em cache para o disco. Observe que alguns SO realizarão uma transferência de dados completa, mas alguns outros apenas fazem uma tentativa de "melhor esforço". Existem três configurações: no: fsync() nunca é chamado. O SO liberará os dados quando estiver pronto. Este modo oferece o mais alto desempenho. sempre: fsync() é chamado após cada gravação no AOF. Este modo é muito lento, mas também muito seguro. Everysec: fsync() é chamado uma vez por segundo. Este modo proporciona um compromisso entre segurança e desempenho.	● não ● Sempre ● a cada seg	a cada seg

Parameter	Descrição	Value Range	Default Value
apenas anexação	Indica se cada modificação da instância deve ou não ser registrada. Por padrão, dados são gravados em discos de maneira assíncrona no Redis. Se essa função estiver desativada, os dados gerados recentemente poderão ser perdidos no caso de uma falha de energia. Opções: **yes**: Os logs são ativados, ou seja, a persistência é ativada. **no**: Os logs são desabilitados, ou seja, a persistência é desabilitada.	● Sim ● não	Sim
client-output-buffer-limit-slave-soft-segundos	Número de segundos que o buffer de saída permanece acima do **client-output-buffer-slave-soft-limit** antes que o cliente seja desconectado.	0–60	60
client-output-buffer-slave-hard-limit	Limite rígido (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite rígido, o cliente é imediatamente desconectado.	0–17.179.869.184	1.717.986.918
client-output-buffer-slave-soft-limit	Limite suave (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite suave e permanece continuamente acima do limite pelo tempo especificado pelo parâmetro **client-output-buffer-limit-slave-soft-seconds**, o cliente é desconectado.	0–17.179.869.184	1.717.986.918

Parameter	Descrição	Value Range	Default Value
política de maxmemória	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	● volatile-lru ● allkeys-lru ● volatile-random ● allkeys-random ● volatile-ttl ● noeviction	volatile-lru **NOTA** Se a instância do DCS Redis for criada antes de julho de 2020 e esse parâmetro não tiver sido modificado, o valor padrão será **noeviction**. Se a instância for criada depois de julho de 2020, o valor padrão será **volatile-lru**.
lua-limite de tempo	Tempo máximo permitido para executar um script Lua (em milissegundos).	100–5000	5000
mestre-somente-leitura	Define a instância como somente leitura. Todas as operações de escrita falharão.	● Sim ● não	não
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–50.000	10.000
proto-max-bulk-len	Tamanho máximo de uma solicitação de um único elemento (em bytes).	1.048.576–536.870.912	536.870.912

Parameter	Descrição	Value Range	Default Value
repl-backlog-tamanho	O tamanho do backlog de replicação (bytes). O backlog é um buffer que acumula dados de réplica quando réplicas são desconectadas do principal. Quando uma réplica é reconectada, uma sincronização parcial é realizada para sincronizar os dados que foram perdidos enquanto as réplicas eram desconectadas.	16.384–1.073.741.824	1.048.576
repl-backlog-ttl	A quantidade de tempo, em segundos, antes do buffer de backlog ser liberado, a partir da última vez que uma réplica foi desconectada. O valor **0** indica que o backlog nunca é liberado.	0–604.800	3600
repl- timeout	Tempo limite de replicação (em segundos).	30–3600	60
hash-max-ziplist-entradas	O número máximo de hashes que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	512
hash-max-ziplist-value	O maior valor permitido para um hash codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
set-max-intset-entradas	Se um conjunto é composto inteiramente de cadeias de caracteres que são inteiros em radix 10 dentro do intervalo de inteiros com sinal de 64 bits, o conjunto é codificado usando intset, uma estrutura de dados otimizada para uso de memória.	1–10.000	512

Parameter	Descrição	Value Range	Default Value
zset-max-ziplist-entradas	O número máximo de conjuntos classificados que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	128
zset-max-ziplist-valor	O maior valor permitido para um conjunto ordenado codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
latência-monitor-limiar	A quantidade mínima de latência que será registrada como picos de latência ● configure para 0: O monitoramento de latência está desativado. ● Definir como mais de 0: Tudo com pelo menos este tempo de latência (em ms) será registrado. Ao executar o comando **LATENCY**, você pode executar operações relacionadas ao monitoramento de latência, como obter dados estatísticos e configurar e ativar o monitoramento de latência.	0–86.400.000 ms	0

Parameter	Descrição	Value Range	Default Value
notific-keyspace-events	Controla para quais notificações de eventos de keyspace são ativadas. Se esse parâmetro estiver configurado, o recurso Redis Pub/Sub permitirá que os clientes recebam uma notificação de evento quando um conjunto de dados do Redis for modificado. As instâncias de cluster de proxy não têm esse parâmetro.	Uma combinação de valores diferentes pode ser usada para ativar notificações para vários tipos de eventos. Os valores possíveis incluem: K: Eventos de espaço de chave, publicados com o prefixo __keyspace@__ e: Eventos keyevent, publicados com o prefixo __keyevent@__ g: Comandos genéricos (não específicos do tipo), como DEL, EXPIRE e RENAME $: Comandos de string eu: Comandos de lista s: Definir comandos h: Comandos de hash z: Comandos do conjunto classificado x: Eventos expirados (eventos gerados toda vez que uma chave expira) e: Eventos despejados (eventos gerados quando uma chave é despejada da maxmemory) Para obter mais informações, consulte a seguinte nota.	Ex
slowlog-log-mais lento-do que	A quantidade máxima de tempo permitida, em microssegundos, para execução de comandos. Se esse limite for excedido, o log de consultas lentas do Redis registrará o comando.	0–1.000.000	10.000

Parameter	Descrição	Value Range	Default Value
slowlog-max-len	O número máximo permitido de consultas lentas que podem ser registradas. O log de consulta lento consome memória, mas você pode recuperar essa memória executando o comando **SLOWLOG RESET**.	0–1000	128

NOTA

1. Os valores padrão e intervalos de valores do **maxclients**, **reserved-memory-percent**, **client-output-buffer-slave-soft-limit**, e os parâmetros **client-output-buffer-slave-hard-limit** estão relacionados às especificações da instância. Portanto, esses parâmetros não são exibidos no modelo de parâmetro.
2. Para obter mais informações sobre os parâmetros descritos em **Tabela 10-1**, visite **https://redis.io/topics/memory-optimization**.

----Fim

10.2 Criando um Modelo de Parâmetro Personalizado

Você pode criar modelos de parâmetros personalizados para diferentes versões do mecanismo de cache e tipos de instância com base nos requisitos de serviço.

Procedimento

Passo 1 Efetue login no console do DCS.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Parameter Templates**.

Passo 4 Clique na guia **Default Templates** ou **Custom Templates** para criar um modelo baseado em um modelo padrão ou em um modelo personalizado existente.

- Se você selecionar **Default Templates**, clique em **Customize** na coluna **Operation** da linha que contém a versão do mecanismo de cache desejada.
- Se você selecionar **Custom Templates**, clique em **Copy** na coluna **Operation** na linha que contém o modelo personalizado desejado.

Passo 5 Especifique **Template Name** e **Description**.

NOTA

O nome do modelo pode conter de 4 a 64 caracteres e deve começar com uma letra ou dígito. Apenas letras, dígitos, hifens (-), sublinhados (_) e pontos (.) são permitidos. A descrição pode estar vazia.

Passo 6 Selecione **Modifiable parameters**.

Atualmente, você pode inserir uma palavra-chave na caixa de pesquisa para pesquisar um parâmetro pelo nome do parâmetro.

Passo 7 Na linha que contém o parâmetro a ser modificado, insira um valor na coluna **Assigned Value**.

Tabela 10-2 descreve os parâmetros. Na maioria dos casos, os valores padrão são mantidos.

Tabela 10-2 Parâmetros de configuração da instância do DCS Redis

Parameter	Descrição	Value Range	Default Value
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0
apêndicefsync	Controla a frequência com que o fsync() transfere dados em cache para o disco. Observe que alguns SO realizarão uma transferência de dados completa, mas alguns outros apenas fazem uma tentativa de "melhor esforço". Existem três configurações: no: fsync() nunca é chamado. O SO liberará os dados quando estiver pronto. Este modo oferece o mais alto desempenho. sempre: fsync() é chamado após cada gravação no AOF. Este modo é muito lento, mas também muito seguro. Everysec: fsync() é chamado uma vez por segundo. Este modo proporciona um compromisso entre segurança e desempenho.	● não ● Sempre ● a cada seg	a cada seg

Parameter	Descrição	Value Range	Default Value
apenas anexação	Indica se cada modificação da instância deve ou não ser registrada. Por padrão, dados são gravados em discos de maneira assíncrona no Redis. Se essa função estiver desativada, os dados gerados recentemente poderão ser perdidos no caso de uma falha de energia. Opções: **yes**: Os logs são ativados, ou seja, a persistência é ativada. **no**: Os logs são desabilitados, ou seja, a persistência é desabilitada.	● Sim ● não	Sim
client-output-buffer-limit-slave-soft-segundos	Número de segundos que o buffer de saída permanece acima do **client-output-buffer-slave-soft-limit** antes que o cliente seja desconectado.	0–60	60
client-output-buffer-slave-hard-limit	Limite rígido (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite rígido, o cliente é imediatamente desconectado.	0–17.179.869.184	1.717.986.918
client-output-buffer-slave-soft-limit	Limite suave (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite suave e permanece continuamente acima do limite pelo tempo especificado pelo parâmetro **client-output-buffer-limit-slave-soft-seconds**, o cliente é desconectado.	0–17.179.869.184	1.717.986.918

Parameter	Descrição	Value Range	Default Value
política de maxmemória	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	● volatile-lru ● allkeys-lru ● volatile-random ● allkeys-random ● volatile-ttl ● noeviction	volatile-lru **NOTA** Se a instância do DCS Redis for criada antes de julho de 2020 e esse parâmetro não tiver sido modificado, o valor padrão será **noeviction**. Se a instância for criada depois de julho de 2020, o valor padrão será **volatile-lru**.
lua-limite de tempo	Tempo máximo permitido para executar um script Lua (em milissegundos).	100–5000	5000
mestre-somente-leitura	Define a instância como somente leitura. Todas as operações de escrita falharão.	● Sim ● não	não
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–50.000	10.000
proto-max-bulk-len	Tamanho máximo de uma solicitação de um único elemento (em bytes).	1.048.576–536.870.912	536.870.912

Parameter	Descrição	Value Range	Default Value
repl-backlog-tamanho	O tamanho do backlog de replicação (bytes). O backlog é um buffer que acumula dados de réplica quando réplicas são desconectadas do principal. Quando uma réplica é reconectada, uma sincronização parcial é realizada para sincronizar os dados que foram perdidos enquanto as réplicas eram desconectadas.	16.384–1.073.741.824	1.048.576
repl-backlog-ttl	A quantidade de tempo, em segundos, antes do buffer de backlog ser liberado, a partir da última vez que uma réplica foi desconectada. O valor **0** indica que o backlog nunca é liberado.	0–604.800	3600
repl- timeout	Tempo limite de replicação (em segundos).	30–3600	60
hash-max-ziplist-entradas	O número máximo de hashes que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	512
hash-max-ziplist-value	O maior valor permitido para um hash codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
set-max-intset-entradas	Se um conjunto é composto inteiramente de cadeias de caracteres que são inteiros em radix 10 dentro do intervalo de inteiros com sinal de 64 bits, o conjunto é codificado usando intset, uma estrutura de dados otimizada para uso de memória.	1–10.000	512

Parameter	Descrição	Value Range	Default Value
zset-max-ziplist-entradas	O número máximo de conjuntos classificados que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	128
zset-max-ziplist-valor	O maior valor permitido para um conjunto ordenado codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
latência-monitor-limiar	A quantidade mínima de latência que será registrada como picos de latência ● configure para 0: O monitoramento de latência está desativado. ● Definir como mais de 0: Tudo com pelo menos este tempo de latência (em ms) será registrado. Ao executar o comando **LATENCY**, você pode executar operações relacionadas ao monitoramento de latência, como obter dados estatísticos e configurar e ativar o monitoramento de latência.	0–86.400.000 ms	0

Parameter	Descrição	Value Range	Default Value
notific-keyspace-events	Controla para quais notificações de eventos de keyspace são ativadas. Se esse parâmetro estiver configurado, o recurso Redis Pub/Sub permitirá que os clientes recebam uma notificação de evento quando um conjunto de dados do Redis for modificado. As instâncias de cluster de proxy não têm esse parâmetro.	Uma combinação de valores diferentes pode ser usada para ativar notificações para vários tipos de eventos. Os valores possíveis incluem: K: Eventos de espaço de chave, publicados com o prefixo __keyspace@__ e: Eventos keyevent, publicados com o prefixo __keyevent@__ g: Comandos genéricos (não específicos do tipo), como DEL, EXPIRE e RENAME $: Comandos de string eu: Comandos de lista s: Definir comandos h: Comandos de hash z: Comandos do conjunto classificado x: Eventos expirados (eventos gerados toda vez que uma chave expira) e: Eventos despejados (eventos gerados quando uma chave é despejada da maxmemory) Para obter mais informações, consulte a seguinte nota.	Ex
slowlog-log-mais lento-do que	A quantidade máxima de tempo permitida, em microssegundos, para execução de comandos. Se esse limite for excedido, o log de consultas lentas do Redis registrará o comando.	0–1.000.000	10.000

Parameter	Descrição	Value Range	Default Value
slowlog-max-len	O número máximo permitido de consultas lentas que podem ser registradas. O log de consulta lento consome memória, mas você pode recuperar essa memória executando o comando **SLOWLOG RESET**.	0–1000	128

NOTA

1. Os valores padrão e intervalos de valores do **maxclients**, **reserved-memory-percent**, **client-output-buffer-slave-soft-limit**, e os parâmetros **client-output-buffer-slave-hard-limit** estão relacionados às especificações da instância. Portanto, esses parâmetros não podem ser modificados.
2. Para obter mais informações sobre os parâmetros descritos em **Tabela 10-2**, visite **https://redis.io/topics/memory-optimization**.
3. O parâmetro **latency-monitor-threshold** é normalmente usado para localização de falhas. Depois de localizar falhas com base nas informações de latência coletadas, altere o valor de **latency-monitor-threshold** para **0** para evitar latência desnecessária.
4. Mais informações sobre o parâmetro **notify-keyspace-events**:
   - A configuração do parâmetro deve conter pelo menos um **K** ou **E**.
   - **A** é um apelido para "g$lshzxe" e não pode ser usado junto com qualquer um dos caracteres em "g$lshzxe".
   - Por exemplo, o valor **Kl** significa que o Redis pode notificar clientes Pub/Sub sobre eventos de espaço de chaves e comandos de lista. O valor **AKE** significa que o Redis notificará os clientes do Pub/Sub sobre todos os eventos.

Passo 8 Clique em **OK**.

----Fim

10.3 Modificando um Modelo de Parâmetro Personalizado

Você pode modificar o nome, a descrição e os parâmetros de um modelo de parâmetro personalizado com base nos requisitos de serviço.

Procedimento

Passo 1 Efetue login no console do DCS.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Parameter Templates**.

Passo 4 Escolha a guia **Custom Templates**.

Passo 5 Você pode modificar um modelo de parâmetro personalizado de uma das seguintes maneiras:

- Clique em **Edit** na coluna **Operation**.
  a. Altere o nome ou modifique a descrição de um modelo.
  b. Na área **Parameters**, selecione **Modifiable parameters**. Na linha que contém o parâmetro a ser modificado, insira um valor na coluna **Assigned Value**. **Tabela 10-3** descreve os parâmetros. Na maioria dos casos, os valores padrão são mantidos.
  c. Clique em **OK**.
- Clique no nome de um modelo personalizado. Na página exibida, modifique os parâmetros.
  a. Selecione **Modifiable parameters**. Digite uma palavra-chave na caixa de pesquisa para pesquisar um parâmetro pelo nome do parâmetro.
  b. Clique em **Modify**.
  c. Na linha que contém o parâmetro a ser modificado, insira um valor na coluna **Assigned Value**. **Tabela 10-3** descreve os parâmetros. Na maioria dos casos, os valores padrão são mantidos.
  d. Clique em **Save**.

Tabela 10-3 Parâmetros de configuração da instância do DCS Redis

Parameter	Descrição	Value Range	Default Value
tempo limite	O período máximo de tempo (em segundos) que uma conexão entre um cliente e a instância do DCS pode permanecer ociosa antes que a conexão seja encerrada. Uma configuração de **0** significa que esta função está desabilitada.	0–7200 segundos	0

Parameter	Descrição	Value Range	Default Value
apêndicefsync	Controla a frequência com que o fsync() transfere dados em cache para o disco. Observe que alguns SO realizarão uma transferência de dados completa, mas alguns outros apenas fazem uma tentativa de "melhor esforço". Existem três configurações: no: fsync() nunca é chamado. O SO liberará os dados quando estiver pronto. Este modo oferece o mais alto desempenho. sempre: fsync() é chamado após cada gravação no AOF. Este modo é muito lento, mas também muito seguro. Everysec: fsync() é chamado uma vez por segundo. Este modo proporciona um compromisso entre segurança e desempenho.	● não ● Sempre ● a cada seg	a cada seg
apenas anexação	Indica se cada modificação da instância deve ou não ser registrada. Por padrão, dados são gravados em discos de maneira assíncrona no Redis. Se essa função estiver desativada, os dados gerados recentemente poderão ser perdidos no caso de uma falha de energia. Opções: **yes**: Os logs são ativados, ou seja, a persistência é ativada. **no**: Os logs são desabilitados, ou seja, a persistência é desabilitada.	● Sim ● não	Sim

Parameter	Descrição	Value Range	Default Value
client-output-buffer-limit-slave-soft-segundos	Número de segundos que o buffer de saída permanece acima do **client-output-buffer-slave-soft-limit** antes que o cliente seja desconectado.	0–60	60
client-output-buffer-slave-hard-limit	Limite rígido (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite rígido, o cliente é imediatamente desconectado.	0–17.179.869.184	1.717.986.918
client-output-buffer-slave-soft-limit	Limite suave (em bytes) no buffer de saída de clientes de réplica. Uma vez que o buffer de saída excede o limite suave e permanece continuamente acima do limite pelo tempo especificado pelo parâmetro **client-output-buffer-limit-slave-soft-seconds**, o cliente é desconectado.	0–17.179.869.184	1.717.986.918

Parameter	Descrição	Value Range	Default Value
política de maxmemória	A política aplicada quando o limite maxmemory é atingido. Para obter mais informações sobre esse parâmetro, consulte **https://redis.io/topics/lru-cache**.	● volatile-lru ● allkeys-lru ● volatile-random ● allkeys-random ● volatile-ttl ● noeviction	volatile-lru **NOTA** Se a instância do DCS Redis for criada antes de julho de 2020 e esse parâmetro não tiver sido modificado, o valor padrão será **noeviction**. Se a instância for criada depois de julho de 2020, o valor padrão será **volatile-lru**.
lua-limite de tempo	Tempo máximo permitido para executar um script Lua (em milissegundos).	100–5000	5000
mestre-somente-leitura	Define a instância como somente leitura. Todas as operações de escrita falharão.	● Sim ● não	não
maxclientes	O número máximo de clientes que podem ser conectados simultaneamente a uma instância de DCS.	1000–50.000	10.000
proto-max-bulk-len	Tamanho máximo de uma solicitação de um único elemento (em bytes).	1.048.576–536.870.912	536.870.912

Parameter	Descrição	Value Range	Default Value
repl-backlog-tamanho	O tamanho do backlog de replicação (bytes). O backlog é um buffer que acumula dados de réplica quando réplicas são desconectadas do principal. Quando uma réplica é reconectada, uma sincronização parcial é realizada para sincronizar os dados que foram perdidos enquanto as réplicas eram desconectadas.	16.384–1.073.741.824	1.048.576
repl-backlog-ttl	A quantidade de tempo, em segundos, antes do buffer de backlog ser liberado, a partir da última vez que uma réplica foi desconectada. O valor **0** indica que o backlog nunca é liberado.	0–604.800	3600
repl- timeout	Tempo limite de replicação (em segundos).	30–3600	60
hash-max-ziplist-entradas	O número máximo de hashes que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	512
hash-max-ziplist-value	O maior valor permitido para um hash codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
set-max-intset-entradas	Se um conjunto é composto inteiramente de cadeias de caracteres que são inteiros em radix 10 dentro do intervalo de inteiros com sinal de 64 bits, o conjunto é codificado usando intset, uma estrutura de dados otimizada para uso de memória.	1–10.000	512

Parameter	Descrição	Value Range	Default Value
zset-max-ziplist-entradas	O número máximo de conjuntos classificados que podem ser codificados usando ziplist, uma estrutura de dados otimizada para reduzir o uso de memória.	1–10.000	128
zset-max-ziplist-valor	O maior valor permitido para um conjunto ordenado codificado usando ziplist, uma estrutura de dados especial otimizada para uso de memória.	1–10.000	64
latência-monitor-limiar	A quantidade mínima de latência que será registrada como picos de latência ● configure para 0: O monitoramento de latência está desativado. ● Definir como mais de 0: Tudo com pelo menos este tempo de latência (em ms) será registrado. Ao executar o comando **LATENCY**, você pode executar operações relacionadas ao monitoramento de latência, como obter dados estatísticos e configurar e ativar o monitoramento de latência.	0–86.400.000 ms	0

Parameter	Descrição	Value Range	Default Value
notific-keyspace-events	Controla para quais notificações de eventos de keyspace são ativadas. Se esse parâmetro estiver configurado, o recurso Redis Pub/Sub permitirá que os clientes recebam uma notificação de evento quando um conjunto de dados do Redis for modificado. As instâncias de cluster de proxy não têm esse parâmetro.	Uma combinação de valores diferentes pode ser usada para ativar notificações para vários tipos de eventos. Os valores possíveis incluem: K: Eventos de espaço de chave, publicados com o prefixo __keyspace@__ e: Eventos keyevent, publicados com o prefixo __keyevent@__ g: Comandos genéricos (não específicos do tipo), como DEL, EXPIRE e RENAME $: Comandos de string eu: Comandos de lista s: Definir comandos h: Comandos de hash z: Comandos do conjunto classificado x: Eventos expirados (eventos gerados toda vez que uma chave expira) e: Eventos despejados (eventos gerados quando uma chave é despejada da maxmemory) Para obter mais informações, consulte a seguinte nota.	Ex
slowlog-log-mais lento-do que	A quantidade máxima de tempo permitida, em microssegundos, para execução de comandos. Se esse limite for excedido, o log de consultas lentas do Redis registrará o comando.	0–1.000.000	10.000

Parameter	Descrição	Value Range	Default Value
slowlog-max-len	O número máximo permitido de consultas lentas que podem ser registradas. O log de consulta lento consome memória, mas você pode recuperar essa memória executando o comando **SLOWLOG RESET**.	0–1000	128

NOTA

1. Os valores padrão e intervalos de valores do **maxclients**, **reserved-memory-percent**, **client-output-buffer-slave-soft-limit**, e os parâmetros **client-output-buffer-slave-hard-limit** estão relacionados às especificações da instância. Portanto, esses parâmetros não podem ser modificados.
2. Para obter mais informações sobre os parâmetros descritos em **Tabela 10-3**, visite **https://redis.io/topics/memory-optimization**.
3. O parâmetro **latency-monitor-threshold** é normalmente usado para localização de falhas. Depois de localizar falhas com base nas informações de latência coletadas, altere o valor de **latency-monitor-threshold** para **0** para evitar latência desnecessária.
4. Mais informações sobre o parâmetro **notify-keyspace-events**:
   - A configuração do parâmetro deve conter pelo menos um **K** ou **E**.
   - **A** é um apelido para "g$lshzxe" e não pode ser usado junto com qualquer um dos caracteres em "g$lshzxe".
   - Por exemplo, o valor **Kl** significa que o Redis pode notificar clientes Pub/Sub sobre eventos de espaço de chaves e comandos de lista. O valor **AKE** significa que o Redis notificará os clientes do Pub/Sub sobre todos os eventos.

----Fim

10.4 Excluindo um Modelo de Parâmetro Personalizado

Esta seção descreve como excluir um modelo de parâmetro personalizado.

Procedimento

Passo 1 Efetue login no console do DCS.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Parameter Templates**.

Passo 4 Escolha a guia **Custom Templates**.

Passo 5 Clique em **Delete** na coluna **Operation**.

Passo 6 Clique em **Yes**.

----Fim

11 Gestão de senhas

11.1 Senhas de instância do DCS

As senhas podem ser configuradas para controlar o acesso às instâncias do DCS, garantindo a segurança dos seus dados.

Você pode definir uma senha durante ou após a criação da instância. Para obter detalhes sobre como definir uma senha após a criação de uma instância, consulte **Redefinindo senhas de instância**.

Você pode escolher se deseja habilitar o acesso sem senha com base na sua segurança e conveniência trade-off.

Cenários que exigem senhas

- Para uma instância de DCS usada na rede ativa ou que contenha informações importantes, é aconselhável definir uma senha.
- Para uma instância de DCS com acesso público habilitado, uma senha deve ser definida para garantir a segurança dos dados.

Para obter detalhes sobre como acessar uma instância com uma senha, consulte **Como acessar uma instância DCS**.

Usando senhas com segurança

1. Ocultar a senha ao usar o redis-cli.

   Se a opção **-a <password>** for usada no redis-cli no Linux, a senha estará sujeita a vazamentos porque é registrada e mantida no histórico. Você é aconselhado a não usar a opção **-a <password>** ao executar comandos no redis-cli. Depois de se conectar ao Redis, execute o comando **auth** para concluir a autenticação, conforme mostrado no exemplo a seguir:

   ```
   $ redis-cli -h 192.168.0.148 -p 6379
   redis 192.168.0.148:6379>auth yourPassword
   OK
   redis 192.168.0.148:6379>
   ```

2. Use a autenticação de senha interativa ou alterne entre usuários com permissões diferentes.

Se o script envolver o acesso à instância do DCS, use a autenticação de senha interativa. Para habilitar a execução automática do script, gerencie o script como outro usuário e autorize a execução usando o sudo.

3. Use um módulo de criptografia em seu aplicativo para criptografar a senha.

11.2 Alteração de senhas de instância

No console do DCS, você pode alterar a senha necessária para acessar a instância do DCS.

NOTA

- Você não pode alterar a senha de uma instância de DCS no modo sem senha.
- A instância de DCS para a qual você deseja alterar a senha está no estado **Running**.
- A nova senha entra em vigor imediatamente no servidor sem a necessidade de reinicialização. O cliente deve se reconectar ao servidor usando a nova senha depois que uma conexão pconnect for fechada. (A senha antiga ainda pode ser usada antes da desconexão.)

Pré-requisitos

Uma instância de DCS foi criada.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Escolha **More** > **Change Password** na linha que contém a instância escolhida.

Passo 5 Na caixa de diálogo exibida, defina **Old Password**, **New Password** e **Confirm Password**.

NOTA

Após 5 tentativas consecutivas de senha incorreta, a conta para acessar a instância DCS escolhida será bloqueada por 5 minutos. As senhas não podem ser alteradas durante o período de bloqueio.

A senha deve atender aos seguintes requisitos:

- Não pode ser deixado em branco.
- Não pode ser igual a senha anterior.
- Pode ter de 8 a 64 caracteres.
- Contenha pelo menos três dos seguintes tipos de caracteres:
  - Letras minúsculas
  - Letras maiúsculas
  - Dígitos
  - caracteres especiais (`~!@#$^&*()-_=+\|{}:,<.>/?)

Passo 6 Na caixa de diálogo **Change Password**, clique em **OK** para confirmar a alteração da senha.

----Fim

11.3 Redefinindo senhas de instância

No console do DCS, você pode configurar uma nova senha se esquecer a senha da instância.

NOTA

- Para uma instância do DCS Redis ou Memcached, você pode alterá-la do modo de senha para o modo sem senha ou do modo sem senha para o modo senha, redefinindo sua senha. Para mais detalhes, consulte **Alterando as Configurações de Senha para Instâncias de Memcached do DCS**.
- A instância DCS para a qual você deseja redefinir a senha está no estado **Running**.

Pré-requisitos

Uma instância de DCS foi criada.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Escolha **More** > **Reset Password** na linha que contém a instância escolhida.

Passo 5 Na caixa de diálogo exibida, defina **New Password** e **Confirm Password**.

NOTA

A senha deve atender aos seguintes requisitos:

- Não pode ser deixado em branco.
- Pode ter de 8 a 64 caracteres.
- Contenha pelo menos três dos seguintes tipos de caracteres:
  - Letras minúsculas
  - Letras maiúsculas
  - Dígitos
  - caracteres especiais (`~!@#$^&*()-_=+\|{}:,<.>/?)

Passo 6 Clique em **OK**.

NOTA

O sistema exibirá uma mensagem de sucesso somente depois que a senha for redefinida com êxito em todos os nós. Se a redefinição falhar, a instância será reiniciada e a senha da instância de cache será restaurada.

----Fim

11.4 Alteração das configurações de senha para instâncias do DCS Redis

Cenário

As instâncias do DCS Redis podem ser acessadas com ou sem senhas. Depois que uma instância é criada, você pode alterar sua configuração de senha nos seguintes cenários:

- Para habilitar o acesso público para uma instância do DCS Redis 3.0, altere a instância para o modo protegido por senha antes de habilitar o acesso público.
- Para acessar uma instância do DCS Redis no modo sem senha, você pode ativar o acesso sem senha para limpar a senha existente da instância.

NOTA

- Para alterar a configuração de senha, a instância do DCS Redis deve estar no estado **Running**.
- O acesso sem senha pode comprometer a segurança. Você pode definir uma senha usando a função de redefinição de senha.
- Por motivos de segurança, o acesso sem senha deve ser desativado quando o acesso público estiver habilitado.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Para alterar a configuração de senha de uma instância do DCS Redis, escolha **Operation** > **More** > **Reset Password** na linha que contém a instância escolhida.

Passo 5 Na caixa de diálogo **Reset Password**, execute uma das seguintes operações, conforme necessário:

- De protegido por senha a livre de senha:

  Alterne a alternância para **Password-Free Access** e clique em **OK**.
- De livre de senha a protegido por senha:

  Digite uma senha, confirme a senha e clique em **OK**.

----Fim

11.5 Alterando as Configurações de Senha para Instâncias de Memcached do DCS

Cenário

As instâncias do Memcached DCS podem ser acessadas com ou sem senhas. Depois que uma instância é criada, você pode alterar sua configuração de senha nos seguintes cenários:

- Se quiser acessar uma instância do Memcached DCS protegida por senha sem o nome de usuário e a senha, você pode habilitar o acesso sem senha para limpar o nome de usuário e a senha da instância.

  O protocolo de texto Memcached não suporta autenticação de nome de usuário e senha. Para acessar uma instância do Memcached DCS usando o protocolo de texto Memcached, você deve habilitar o acesso sem senha à instância.
- Se você quiser acessar uma instância do Memcached DCS sem senha usando um nome de usuário e uma senha, defina uma senha para a instância usando a função de redefinição de senha.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Para habilitar o acesso sem senha a uma instância do Memcached DCS, escolha **Operation** > **More** > **Reset Password** na linha que contém a instância escolhida.

Passo 5 Na caixa de diálogo **Reset Password**, execute uma das seguintes operações, conforme necessário:

- De protegido por senha a livre de senha:

  Alterne o toggle para **Password-Free Access** e clique em **OK**.
- De livre de senha a protegido por senha:

  Digite uma senha, confirme a senha e clique em **OK**.

----Fim

12 Cotas

O que é Quota?

Uma cota é um limite na quantidade ou capacidade de um determinado tipo de recursos de serviço que você pode usar, por exemplo, o número máximo de instâncias de DCS que você pode criar e a quantidade máxima de memória que você pode usar.

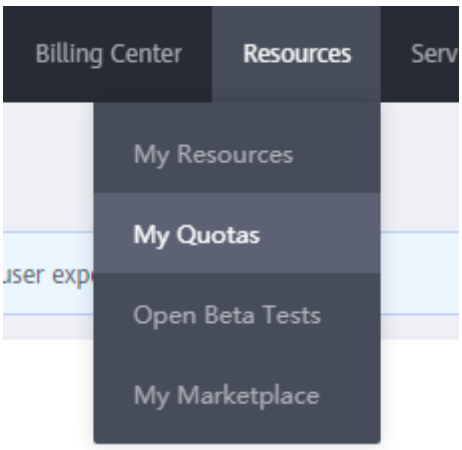
Se uma cota não puder atender às suas necessidades, solicite uma cota mais alta.

Como faço para visualizar minha cota?

1. Acesse o console de gerenciamento.
2. Clique em no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.
3. No canto superior direito da página, escolha **Resources** > **My Quotas**.

   A página **Service Quota** é exibida.

   Figura 12-1 Minha cotas

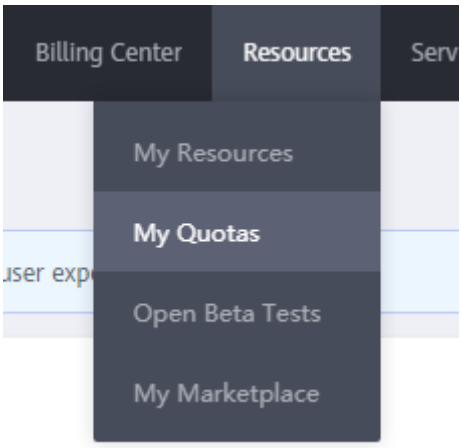
4. Na página **Service Quota**, exiba as cotas usadas e totais de recursos.

   Se uma cota não puder atender às suas necessidades, solicite uma cota mais alta realizando as seguintes operações.

Como faço para aumentar minha cota?

1. Acesse o console de gerenciamento.
2. No canto superior direito da página, escolha **Resources** > **My Quotas**.
   A página **Service Quota** é exibida.

   Figura 12-2 Minha cotas

3. Clique em **Increase Quota**.
4. Na página **Create Service Ticket**, defina os parâmetros.
   Na área **Problem Description**, insira a cota necessária e o motivo do ajuste da cota.
5. Leia os contratos e confirme que concorda com eles e, em seguida, clique em **Submit**.

13 Monitoramento

O Cloud Eye é uma plataforma de monitoramento segura e escalável. Ele monitora as métricas do DCS e envia notificações se os alarmes forem acionados ou ocorrerem eventos.

13.1 Métricas DCS

Introdução

Esta seção descreve as métricas de DCS relatadas ao Cloud Eye, bem como seus namespaces e dimensões. Você pode usar o console do Cloud Eye ou chamar **as API** para consultar as métricas e os alarmes do DCS.

Diferentes tipos de instâncias são monitoradas em diferentes dimensões.

Tabela 13-1 Monitorando dimensões para diferentes tipos de instância

Tipos de instância	Monitoramento de instância	Monitoramento do servidor Redis	Monitoramento de Proxy
Único-nó	Compatível O monitoramento na dimensão de instância é conduzido no Servidor Redis.	N/A	N/A
Principal/Em espera	Compatível O nó principal é monitorado.	Compatível Os nós principais e em espera são monitorados.	N/A
Cluster de proxy	Compatível Os dados de monitoramento os dados de nó principal agregados.	Compatível Cada estilhaço é monitorado.	Compatível Cada proxy é monitorado.

Tipos de instância	Monitoramento de instância	Monitoramento do servidor Redis	Monitoramento de Proxy
Cluster do Redis	Compatível Os dados de monitoramento os dados de nó principal agregados.	Compatível Cada estilhaço é monitorado.	N/A

Espaço de nomes

SYS.DCS

Métricas de instância do DCS Redis 3.0

NOTA

- O DCS for Redis 3.0 não é mais fornecido. Em vez disso, você pode usar o DCS for Redis 4.0 ou 5.0.
- **Dimensões** lista as dimensões métricas.

Tabela 13-2 Métricas de instância do DCS Redis 3.0

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
cpu_usagem	Uso da CPU	Uso máximo da CPU do objeto monitorado entre vários valores de amostragem em um período de monitoramento Unidade: % Para uma instância de nó único ou principal/em espera, essa métrica indica o uso da CPU do nó principal. Para uma instância de Cluster de Proxy, essa métrica indica o valor médio de todos os proxies.	0–100%	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
uso_de_memória	Uso da memória	Memória consumida pelo objeto monitorado Unidade: % **AVISO** O uso de memória não inclui o uso de memória reservada.	0–100%	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
net_in_throughput	Throughput de entrada da rede	throughput de entrada por segundo em uma porta Unidade: byte/s	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
rendimento_de_saída_rede	Throughput de saída da rede	throughput de saída por segundo em uma porta Unidade: byte/s	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
clientes_conectados	Clientes conectados	Número de clientes conectados (excluindo os de nós escravos)	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
lista_de_saída_mais_longa_do_cliente	Lista de saída mais longa do cliente	Lista de saída mais longa entre as conexões atuais do cliente	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
cliente_maior_em_buf	Buf de entrada maior do cliente	Comprimento máximo dos dados de entrada entre as conexões atuais do cliente Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
clientes_bloqueados	Clientes bloqueados	Número de clientes suspensos por operações de bloqueio, como BLPOP, BRPOP e BRPOPLPUSH	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
memória_usada	Memória usada	Número de bytes usados pelo servidor Redis Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
usado_memória_rss	Memória usada RSS	Memória RSS (tamanho de conjunto residente) usada pelo servidor Redis, que é a memória que realmente reside na memória, incluindo toda a memória de pilha e heap, mas não a memória trocada Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
pico_de_memória_usado	Pico da Memória Usada	Memória de pico consumida pelo Redis desde a última inicialização do servidor Redis Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
usado_memória_lua	Memória Lua Usada	Número de bytes usados pelo motor Lua Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
memória_frag_ratio	Relação de fragmentação de memória	Fragmentação de memória atual, que é a razão entre **used_memory_rss**/ **used_memory**.	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
total_conexões_recebidas	Novas Conexões	Número de conexões recebidas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
total_de_comandos_processados	Comandos processados	Número de comandos processados durante o período de monitorização	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
instantâneas_ops	Ops por segundo	Número de comandos processados por segundo	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
total_de_input_bytes	Bytes de entrada de rede	Número de bytes recebidos durante o período de monitoramento Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
total_de_saída_bytes	Bytes de saída de rede	Número de bytes enviados durante o período de monitoramento Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
kbps_input_instantâneos	Fluxo de entrada	Tráfego de entrada instantâneo Unidade: KB/s	≥ 0 KB/s	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
kbps_saída_instantânea	Fluxo de saída	Tráfego de saída instantâneo Unidade: KB/s	≥ 0 KB/s	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
ligações_rejeitadas	Conexões Rejeitadas	Número de conexões que excederam maxclients e foram rejeitadas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
chaves_expiradas	Chaves expiradas	Número de chaves que expiraram e foram excluídas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
chaves_despejadas	Chaves despejadas	Número de chaves que foram despejadas e excluídas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
keyspace_hits	Hits do Keyspace	Número de pesquisas bem-sucedidas de chaves no dicionário principal durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
keyspace_falta	Keyspace Misses	Número de pesquisas falhadas de chaves no dicionário principal durante o período de monitorização	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
pubsub_canais	Canais PubSub	Número de canais Pub/Sub	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
pubsub_padrões	Padrões PubSub	Número de padrões Pub/Sub	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
keyspace_hits_perc	Taxa de hits	Relação entre o número de acertos de cache do Redis e o número de pesquisas. Cálculo: keyspace_hits/(keyspace_hits + keyspace_misses) Unidade: %	0–100%	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
atraso_máximo_de_comando	Latência máxima de comandos	Máxima latência dos comandos Unidade: ms	≥ 0 ms	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
auth_errors	Falhas de autenticação	Número de autenticações falhadas	≥ 0	Instância do Redis DCS de nó único ou principal/em espera	1 minuto
é_lento_log_existir	Logs de consulta lentos	Existência de logs de consulta lentos na instância	● **1**: sim ● **0**: não	Instância do Redis DCS de nó único ou principal/em espera	1 minuto
chaves	Chaves	Número de chaves no Redis	≥ 0	Instância do Redis DCS de nó único ou principal/em espera	1 minuto

Métricas de instância do DCS Redis 4.0/5.0

NOTA

Dimensões lista as dimensões métricas.

Tabela 13-3 Métricas de instância do DCS Redis 4.0/5.0

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
cpu_usagem	Uso da CPU	Uso máximo da CPU do objeto monitorado entre vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Instância do Redis DCS de nó único ou principal/em espera	1 minuto
cpu_avg_usagem	Uso médio da CPU	Uso médio da CPU do objeto monitorado de vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Instância do Redis DCS de nó único ou principal/em espera	1 minuto
atraso_máximo_de_comando	Latência Máxima do Comando	Máxima latência dos comandos Unidade: ms	≥ 0 ms	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
total_conexões_recebidas	Novas Conexões	Número de conexões recebidas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
é_lento_log_existir	Logs de consulta lentos	Existência de logs de consulta lentos na instância	● **1**: sim ● **0**: não	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
uso_de_memória	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
Vencimento	Chaves com uma expiração	Número de chaves com expiração no Redis	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
keyspace_hits_perc	Taxa de hits	Relação entre o número de acertos de cache do Redis e o número de pesquisas. Cálculo: keyspace_hits/(keyspace_hits + keyspace_misses) Unidade: %	0–100%	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
memória_usada	Memória Utilizada	Número total de bytes usados pelo servidor Redis Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
conjunto_de_dados_de_memória_usado	Conjunto de dados de memória usado	Memória do conjunto de dados que o servidor Redis utilizou Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
conjunto_de_dados_de_memória_utilizados_perc	Relação de conjunto de dados de memória usada	Percentual da memória de dados que o Redis usou em relação ao total de memória usada Unidade: %	0–100%	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
usado_memória_rss	Memória Usada RSS	Memória RSS (tamanho de conjunto residente) usada pelo servidor Redis, que é a memória que realmente reside na memória, incluindo toda a memória de pilha e heap, mas não a memória trocada Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
instantâneas_ops	Ops por segundo	Número de comandos processados por segundo	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
keyspace_falta	Keyspace Misses	Número de pesquisas falhadas de chaves no dicionário principal durante o período de monitorização	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
chaves	Chaves	Número de chaves no Redis	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
clientes_bloqueados	Clientes bloqueados	Número de clientes suspensos por operações de bloqueio	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
clientes_conectados	Clientes conectados	Número de clientes conectados (excluindo os de nós escravos)	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
del	Del	Número de comandos **DEL** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
chaves_despejadas	Chaves Despejadas	Número de chaves que foram despejadas e excluídas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
vencimento	EXPIRE	Número de comandos **EXPIRE** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
chaves_expiradas	Chaves Expiradas	Número de chaves que expiraram e foram excluídas durante o período de monitoramento	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
obter	GET	Número de comandos **GET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
hdel	HDEL	Número de comandos **HDEL** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
hget	HGET	Número de comandos **HGET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
hmget	HMGET	Número de comandos **HMGET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
hmset	HMSET	Número de comandos **HMSET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
hset	HSET	Número de comandos **HSET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
kbps_input_instantâneos	Fluxo de entrada	Tráfego de entrada instantâneo Unidade: KB/s	≥ 0 KB/s	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
kbps_saída_instantânea	Fluxo de saída	Tráfego de saída instantâneo Unidade: KB/s	≥ 0 KB/s	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
memória_frag_ratio	Relação de fragmentação de memória	Relação entre RSS de Memória Usada e Memória Usada	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
mget	MGET	Número de comandos **MGET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
mset	MSET	Número de comandos **MSET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
pubsub_canais	Canais PubSub	Número de canais Pub/Sub	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
pubsub_padrões	Padrões PubSub	Número de padrões Pub/Sub	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
definir	SET	Número de comandos **SET** processados por segundo	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
usado_memória_lua	Memória Lua Usada	Número de bytes usados pelo motor Lua Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
pico_de_memória_usado	Pico da Memória Usada	Memória de pico consumida pelo Redis desde a última inicialização do servidor Redis Unidade: byte	≥ 0	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
sadd	Sadd	Número de comandos **SADD** processados por segundo Unidade: Conta/s	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto
smembers	Smembers	Número de comandos **SMEMBERS** processados por segundo Unidade: conta(s)	0–500.000	Instância do Redis de nó único, principal/em espera ou cluster	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
rx_controlado	Tempos de controle de fluxo	Número de tempos de controlo do caudal durante o período de monitorização Se o valor for maior que 0, a largura de banda usada excederá o limite superior e o controle de fluxo será acionado. Unidade: Contagem	≥ 0	Instância do Cluster do Redis	1 minuto
uso_de_largura de banda	Uso de largura de banda	Porcentagem da largura de banda utilizada para o limite máximo de largura de banda	0–200%	Instância do Cluster do Redis	1 minuto
comando_max_rt	Máxima Latência	Atraso máximo de quando o nó recebe comandos até quando responde Unidade: nós	≥ 0	Instância do DCS Redis 4.0/5.0/6.0 de nó único	1 minuto
comando_avg_rt	Latência Média	Atraso médio de quando o nó recebe comandos até quando responde Unidade: nós	≥ 0	Instância do DCS Redis 4.0/5.0/6.0 de nó único	1 minuto

Métricas do servidor Redis de instâncias do DCS Redis

NOTA

- Para instâncias de cluster de proxy, o monitoramento abrange servidores Redis e proxies. Para instâncias do Cluster Redis DCS Redis 4.0 e 5.0 e instâncias principal/em espera, o monitoramento abrange apenas servidores Redis.
- **Dimensões** lista as dimensões métricas.

Tabela 13-4 Métricas do servidor Redis

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
cpu_usagem	Uso da CPU	Uso máximo da CPU do objeto monitorado entre vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
cpu_avg_usagem	Uso médio da CPU	Uso médio da CPU do objeto monitorado de vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
uso_de_memória	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
clientes_conectados	Clientes conectados	Número de clientes conectados (excluindo os de nós escravos)	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
lista_de_saída_mais_longa_do_cliente	Lista de saída mais longa do cliente	Lista de saída mais longa entre as conexões atuais do cliente	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis 3.0 ou 4.0	1 minuto
cliente_maior_em_buf	Buf de entrada maior do cliente	Comprimento máximo dos dados de entrada entre as conexões atuais do cliente Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis 3.0 ou 4.0	1 minuto
clientes_bloqueados	Clientes bloqueados	Número de clientes suspensos por operações de bloqueio, como BLPOP, BRPOP e BRPOPLPUSH	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
memória_usada	Memória Utilizada	Número total de bytes usados pelo servidor Redis Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
usado_memória_rss	Memória Usada RSS	Memória RSS usada pelo servidor Redis, que inclui toda a memória de pilha e heap, mas não a memória trocada Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
pico_de_memória_usado	Pico da Memória Usada	Memória de pico consumida pelo Redis desde a última inicialização do servidor Redis Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
usado_memória_lua	Memória Lua Usada	Número de bytes usados pelo motor Lua Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
memória_frag_ratio	Relação de fragmentação de memória	Fragmentação de memória atual, que é a razão entre **used_memory_rss**/ **used_memory**.	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
total_conexões_recebidas	Novas Conexões	Número de conexões recebidas durante o período de monitoramento	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
total_de_comandos_processados	Comandos processados	Número de comandos processados durante o período de monitorização	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
instantâneas_ops	Ops por segundo	Número de comandos processados por segundo	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
total_de_input_bytes	Bytes de entrada de rede	Número de bytes recebidos durante o período de monitoramento Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
total_de_saída_bytes	Bytes de saída de rede	Número de bytes enviados durante o período de monitoramento Unidade: byte	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
kbps_input_instantâneos	Fluxo de entrada	Tráfego de entrada instantâneo Unidade: KB/S	≥ 0 KB/s	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
kbps_saída_instantânea	Fluxo de saída	Tráfego de saída instantâneo Unidade: KB/s	≥ 0 KB/s	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
ligações_rejeitadas	Conexões Rejeitadas	Número de conexões que excederam maxclients e foram rejeitadas durante o período de monitoramento	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
chaves_expiradas	Chaves Expiradas	Número de chaves que expiraram e foram excluídas durante o período de monitoramento	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
chaves_despejadas	Chaves despejadas	Número de chaves que foram despejadas e excluídas durante o período de monitoramento	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
pubsub_canais	Canais PubSub	Número de canais Pub/Sub	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
pubsub_padrões	Padrões PubSub	Número de padrões Pub/Sub	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis 3.0, 4.0 ou 5.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
keyspace_hits_perc	Taxa de hits	Relação entre o número de acertos de cache do Redis e o número de pesquisas. Cálculo: keyspace_hits/(keyspace_hits + keyspace_misses) Unidade: %	0–100%	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
atraso_máximo_de_comando	Latência Máxima do Comando	Máxima latência dos comandos Unidade: ms	≥ 0 ms	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
é_lento_log_existir	Logs de consulta lentos	Existência de logs de consulta lentos no nó	● **1**: sim ● **0**: não	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto
Chaves	Chaves	Número de chaves no Redis	≥ 0	Servidor Redis de uma instância DCS principal/em espera ou cluster Redis	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
Sadd	Sadd	Número de comandos **SADD** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
Smembers	Smembers	Número de comandos **SMEMBERS** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
ms_repl_deslocamento	Gap de replicação	Lacuna de sincronização de dados entre o principal e a réplica	-	Servidor Redis de réplica de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
del	DEL	Número de comandos **DEL** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
vencimento	EXPIRE	Número de comandos **EXPIRE** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
obter	GET	Número de comandos **GET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
hdel	HDEL	Número de comandos **HDEL** processados por segundo Unidade: Conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
hget	HGET	Número de comandos **HGET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
hmget	HMGET	Número de comandos **HMGET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
hmset	HMSET	Número de comandos **HMSET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
hset	HSET	Número de comandos **HSET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
mget	MGET	Número de comandos **MGET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mset	MSET	Número de comandos **MSET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
definir	SET	Número de comandos **SET** processados por segundo Unidade: conta(s)	0–500.000	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
rx_controlado	Tempos de controle de fluxo	Número de tempos de controlo do caudal durante o período de monitorização Se o valor for maior que 0, a largura de banda usada excederá o limite superior e o controle de fluxo será acionado. Unidade: Contagem	≥ 0	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
uso_de_largura de banda	Uso de largura de banda	Porcentagem da largura de banda utilizada para o limite máximo de largura de banda	0–200%	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
uso_de_conexões	Uso de conexão	Porcentagem do número atual de conexões para o número máximo permitido de conexões Unidade: %	0–100%	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
comando_max_rt	Máxima Latência	Atraso máximo desde quando o nó recebe comandos até quando responde Unidade: nós	≥ 0	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
comando_avg_rt	Latência Média	Atraso médio de quando o nó recebe comandos até quando responde Unidade: nós	≥ 0	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
sincronização_completa	Tempos de sincronização completa	Número total de sincronizações completas desde que o servidor Redis foi iniciado pela última vez	≥ 0	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto
slow_log_counts (contagens_de_logs lentas)	Consultas lentas	Número de vezes que as consultas lentas ocorrem dentro de um período de monitoramento	≥ 0	Servidor Redis de uma instância principal/em espera ou cluster DCS Redis 4.0 ou 5.0 ou uma instância principal/em espera DCS Redis 6.0	1 minuto

Métricas de proxy

NOTA

Dimensões lista as dimensões métricas.

Tabela 13-5 Métricas de proxy de instâncias do Cluster de proxy DCS Redis 3.0

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
cpu_usagem	Uso da CPU	Uso máximo da CPU do objeto monitorado entre vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
uso_de_memória	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
p_clientes_conectados	Clientes conectados	Número de clientes conectados	≥ 0	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
max_rxpck_por_seg	Taxa máx. de recebimento de pacotes de dados NIC	Número máximo de pacotes de dados recebidos pela NIC proxy por segundo durante o período de monitoramento Unidade: pacotes/segundo	0–10.000.000	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
max_txpck_por_seg	Max. Taxa de transmissão de pacotes de dados NIC	Número máximo de pacotes de dados transmitidos pela NIC proxy por segundo durante o período de monitoramento Unidade: pacotes/segundo	0–10.000.000	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
max_rxkB_por_seg	Largura de banda de entrada máxima	Maior volume de dados recebidos pelo NIC proxy por segundo Unidade: KB/s	≥ 0 KB/s	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
max_txkB_por_seg	Largura de banda máxima de saída	Maior volume de dados transmitidos pela NIC proxy por segundo Unidade: KB/s	≥ 0 KB/s	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
avg_rxpck_per_sec	Taxa média de recebimento de pacotes de dados NIC	Número médio de pacotes de dados recebidos pela NIC do proxy por segundo durante o período de monitoramento Unidade: pacotes/segundo	0–10.000.000	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
avg_txpck_por_seg	Taxa média de transmissão de pacotes de dados NIC	Número médio de pacotes de dados transmitidos pela NIC proxy por segundo durante o período de monitoramento Unidade: pacotes/segundo	0–10.000.000	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto
avg_rxkB_per_sec	Largura de banda média de entrada	Volume médio de dados recebidos pela NIC proxy por segundo Unidade: KB/s	≥ 0 KB/s	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
avg_txkB_por_seg	Largura de banda média de saída	Volume médio de dados transmitidos pela NIC proxy por segundo Unidade: KB/s	≥ 0 KB/s	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy	1 minuto

Tabela 13-6 Métricas de proxy de instâncias de cluster de proxy DCS Redis 4.0 ou 5.0

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
status_do_nó	Status do proxy	Indicação se o proxy é normal.	• **0**: Normal • **1**: Anormal	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
cpu_usagem	Uso da CPU	Uso máximo da CPU do objeto monitorado entre vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
uso_de_memória	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
p_clientes_conectados	Clientes conectados	Número de clientes conectados	≥ 0	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
instantâneas_ops	Ops por segundo	Número de comandos processados por segundo	≥ 0	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
kbps_input_instantâneos	Fluxo de entrada	Tráfego de entrada instantâneo Unidade: KB/s	≥ 0 KB/s	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
kbps_saída_instantânea	Fluxo de saída	Tráfego de saída instantâneo Unidade: KB/s	≥ 0 KB/s	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
total_de_input_bytes	Bytes de entrada de rede	Número de bytes recebidos durante o período de monitoramento Unidade: byte	≥ 0	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
total_de_saída_bytes	Bytes de saída de rede	Número de bytes enviados durante o período de monitoramento Unidade: byte	≥ 0	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
uso_de_conexões	Uso de conexão	Porcentagem do número atual de conexões para o número máximo permitido de conexões Unidade: %	0–100%	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
comando_max_rt	Máxima Latência	Atraso máximo desde quando o nó recebe comandos até quando responde Unidade: nós	≥ 0	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto
comando_avg_rt	Latência Média	Atraso médio de quando o nó recebe comandos até quando responde Unidade: nós	≥ 0	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0	1 minuto

Métricas da Instância do Memcached DCS

NOTA

Dimensões lista as dimensões métricas.

Tabela 13-7 Métricas de instância do Memcached DCS

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
cpu_usagem	Uso da CPU	Uso máximo da CPU do objeto monitorado entre vários valores de amostragem em um período de monitoramento Unidade: %	0–100%	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
uso_de_memória	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Instância do Memcached DCS	1 minuto
net_in_throughput	Rendimento de entrada da rede	throughput de entrada por segundo em uma porta Unidade: byte/s	≥ 0	Instância do Memcached DCS	1 minuto
rendimento_de_saída_rede	Throughput de saída da rede	throughput de saída por segundo em uma porta Unidade: byte/s	≥ 0	Instância do Memcached DCS	1 minuto
mc_clientes_conectados	Clientes conectados	Número de clientes conectados (excluindo os de nós escravos)	≥ 0	Instância do Memcached DCS	1 minuto
mc_memória_usada	Memória Utilizada	Número de bytes usados pelo Memcached Unidade: byte	≥ 0	Instância do Memcached DCS	1 minuto
mc_used_memória_rss	Memória Usada RSS	Memória RSS usada que realmente reside na memória, incluindo toda a memória de pilha e heap, mas não a memória trocada Unidade: byte	≥ 0	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mc_used_memória_peak	Pico da Memória Usada	Memória de pico consumida desde a última inicialização do servidor Unidade: byte	≥ 0	Instância do Memcached DCS	1 minuto
mc_memória_frag_ratio	Relação de fragmentação de memória	Relação entre RSS de Memória Usada e Memória Usada	≥ 0	Instância do Memcached DCS	1 minuto
mc_conexões_recebidas	Novas Conexões	Número de conexões recebidas durante o período de monitoramento	≥ 0	Instância do Memcached DCS	1 minuto
mc_comandos_processados	Comandos processados	Número de comandos processados durante o período de monitorização	≥ 0	Instância do Memcached DCS	1 minuto
mc_instantâneo_ops	Ops por segundo	Número de comandos processados por segundo	≥ 0	Instância do Memcached DCS	1 minuto
mc_net_input_bytes	Bytes de entrada de rede	Número de bytes recebidos durante o período de monitoramento Unidade: byte	≥ 0	Instância do Memcached DCS	1 minuto
mc_net_output_bytes	Bytes de saída de rede	Número de bytes enviados durante o período de monitoramento Unidade: byte	≥ 0	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mc_instantâneo_input_kbps	Fluxo de entrada	Tráfego de entrada instantâneo Unidade: KB/s	≥ 0 KB/s	Instância do Memcached DCS	1 minuto
mc_saída_instantânea_kbps	Fluxo de saída	Tráfego de saída instantâneo Unidade: KB/s	≥ 0 KB/s	Instância do Memcached DCS	1 minuto
mc_conexões_rejeitadas	Conexões Rejeitadas	Número de conexões que excederam maxclients e foram rejeitadas durante o período de monitoramento	≥ 0	Instância do Memcached DCS	1 minuto
mc_expired_keys	Chaves Expiradas	Número de chaves que expiraram e foram excluídas durante o período de monitoramento	≥ 0	Instância do Memcached DCS	1 minuto
mc_evicted_keys	Chaves despejadas	Número de chaves que foram despejadas e excluídas durante o período de monitoramento	≥ 0	Instância do Memcached DCS	1 minuto
mc_cmd_get	Número de solicitações de recuperação	Número de solicitações de recuperação de dados recebidas	≥ 0	Instância do Memcached DCS	1 minuto
mc_cmd_set	Número de solicitações de armazenamento	Número de solicitações de armazenamento de dados recebidas	≥ 0	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mc_cmd_flush	Número de Solicitações de Flush	Número de solicitações de liberação de dados recebidas	≥ 0	Instância do Memcached DCS	1 minuto
mc_cmd_toque	Número de solicitações de toque	Número de pedidos recebidos para alterar o período de validade dos dados	≥ 0	Instância do Memcached DCS	1 minuto
mc_get_hits	Número de Hits de Recuperação	Número de operações de recuperação de dados bem-sucedidas	≥ 0	Instância do Memcached DCS	1 minuto
mc_get_misses	Número de Ausências de Recuperação	Número de operações de recuperação de dados com falha devido à inexistência de chave	≥ 0	Instância do Memcached DCS	1 minuto
mc_excluir_hits	Número de Sucessos de Deleção	Número de operações de exclusão de dados bem-sucedidas	≥ 0	Instância do Memcached DCS	1 minuto
mc_excluir_misses	Número de Apagar Misses	Número de operações de exclusão de dados com falha devido à inexistência de chave	≥ 0	Instância do Memcached DCS	1 minuto
mc_incr_hits	Número de Incrementos Hits	Número de operações de incremento bem-sucedidas	≥ 0	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mc_incr_misses	Número de Ausências de Incremento	Número de operações de incremento com falha devido à inexistência de chave	≥ 0	Instância do Memcached DCS	1 minuto
mc_decr_hits	Número de Acertos de Decremento	Número de operações de decremento bem-sucedidas	≥ 0	Instância do Memcached DCS	1 minuto
mc_decr_misses	Número de Ausências de Decremento	Número de operações de diminuição falhadas devido à inexistência de chave	≥ 0	Instância do Memcached DCS	1 minuto
mc_cas_hits	Número de acertos de CAS	Número de operações CAS bem-sucedidas	≥ 0	Instância do Memcached DCS	1 minuto
mc_cas_misses	Número de faltas de CAS	Número de operações CAS com falha devido à inexistência de chave	≥ 0	Instância do Memcached DCS	1 minuto
mc_cas_badval	Número de valores CAS não correspondidos	Número de operações CAS falhadas devido à incompatibilidade de valores CAS	≥ 0	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mc_touch_hits	Número de hits do toque	Número de solicitações bem-sucedidas para modificar o período de validade dos dados	≥ 0	Instância do Memcached DCS	1 minuto
mc_touch_misses	Número de faltas de toque	Número de solicitações com falha para modificar o período de validade dos dados devido à inexistência de chave	≥ 0	Instância do Memcached DCS	1 minuto
mc_auth_cmds	Solicitações de autenticação	Número de solicitações de autenticação	≥ 0	Instância do Memcached DCS	1 minuto
mc_auth_errors	Falhas de autenticação	Número de solicitações de autenticação com falha	≥ 0	Instância do Memcached DCS	1 minuto
mc_curr_items	Número de itens armazenados	Número de itens de dados armazenados	≥ 0	Instância do Memcached DCS	1 minuto
mc_comando_max_delay	Latência Máxima do Comando	Máxima latência dos comandos Unidade: ms	≥ 0 ms	Instância do Memcached DCS	1 minuto
mc_é_lento_log_existir	Logs de consulta lentos	Existência de logs de consulta lentos na instância	• **1**: sim • **0**: não	Instância do Memcached DCS	1 minuto

ID da métrica	Nome da métrica	Descrição da Métrica	Intervalo de valores	Objeto monitorado	Período de monitorização (dados brutos)
mc_keyspace_hits_perc	Taxa de hits	Proporção do número de hits de cache do Memcached para o número de pesquisas Unidade: %	0–100%	Instância do Memcached DCS	1 minuto

Dimensões

Chave	Valor
dcs_instance_id	Instância do DCS Redis
dcs_cluster_redis_node	Servidor Redis
dcs_cluster_proxy_node	Proxy em uma instância do DCS Redis 3.0 de cluster de proxy
dcs_cluster_proxy2_node	Proxy em um cluster de proxy Instância do DCS Redis 4.0 ou 5.0
dcs_memcached_instance_id	Instância Memcached do DCS

13.2 Métricas comuns

Esta seção descreve métricas comuns do Redis.

Tabela 13-8 Métricas comuns

Métrica	Descrição
Uso da CPU	Esta métrica indica o valor máximo em cada período de medição (minuto-nível: a cada minuto; segundo nível: a cada 5 segundos). ● Para uma instância de nó único ou principal/em espera, você pode visualizar o uso da CPU da instância. ● Para uma instância de Cluster de Proxy, você pode exibir o uso da CPU dos servidores Redis e dos proxies. ● Para uma instância de cluster do Redis, você só pode exibir o uso da CPU dos servidores Redis.
Uso da memória	Essa métrica mede o uso da memória em cada período de medição (nível de minuto: a cada minuto; segundo nível: a cada 5 segundos). ● Para uma instância de nó único ou principal/em espera, você pode visualizar o uso de memória da instância. ● Para uma instância de Cluster de Proxy, você pode exibir o uso de memória da instância e dos proxies. ● Para uma instância de cluster do Redis, você só pode exibir o uso de memória dos servidores Redis. **AVISO** O uso de memória não inclui o uso de memória reservada.
Clientes conectados	Essa métrica indica o número de clientes conectados instantâneos, ou seja, o número de conexões simultâneas. Essa métrica não inclui o número de conexões aos nós em espera de instâncias principal/em espera ou cluster. Para obter detalhes sobre o número máximo permitido de conexões, consulte o "Máx. Conexões" coluna de diferentes tipos de instância listados em **Especificações da instância do DCS**.
Ops por segundo	Essa métrica indica o número de operações processadas por segundo. Para obter detalhes sobre o número máximo permitido de operações por segundo, consulte a coluna "Reference Performance (QPS)" de diferentes tipos de instância listados em **Especificações da instância do DCS**.
Fluxo de entrada	Essa métrica indica o tráfego de entrada instantâneo. ● Os dados de monitoramento no nível da instância mostram o tráfego de entrada agregado de todos os nós. ● Os dados de monitoramento no nível do nó mostram o tráfego de entrada do nó atual.

Métrica	Descrição
Fluxo de saída	Essa métrica indica o tráfego de saída instantâneo. ● Os dados de monitoramento no nível da instância mostram o tráfego de saída agregado de todos os nós. ● Os dados de monitoramento no nível do nó mostram o tráfego de saída do nó atual.
Uso de largura de banda	Esta métrica indica a porcentagem da largura de banda usada ao limite máximo da largura de banda.
Comandos processados	Essa métrica indica o número de comandos processados durante o período de monitoramento. O período de monitoramento padrão é de 1 minuto. O período de monitoramento dessa métrica é diferente do da métrica **Ops per Second** A métrica **Ops per Second** mede o número instantâneo de comandos processados. A métrica **Commands Processed** mede o número total de comandos processados durante o período de monitoramento.
Tempos de controle de fluxo	Essa métrica indica o número de vezes que a largura de banda máxima permitida é excedida durante o período de monitoramento. Para obter detalhes sobre a largura de banda máxima permitida, consulte a coluna "Largura de banda máxima/garantida" de diferentes tipos de instância listados em **Especificações da instância do DCS**.
Consultas lentas	Essa métrica indica se existem consultas lentas na instância. Para obter detalhes sobre a causa de uma consulta lenta, consulte **Exibindo consultas lentas do Redis**.

13.3 Exibindo Métricas

O serviço Cloud Eye monitora o desempenho em execução das instâncias de DCS.

Procedimento

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em  no canto superior esquerdo do console de gerenciamento e selecione a região onde sua instância está localizada.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Clique na instância desejada.

Passo 5 Escolha **Performance Monitoring**. Todas as métricas de monitoramento da instância são exibidas.

NOTA

Você também pode clicar em **View Metric** na coluna **Operation** da página **Cache Manager**. Você será redirecionado para o console Cloud Eye. As métricas exibidas no console do Cloud Eye são as mesmas exibidas na página **Performance Monitoring** do console do DCS.

----Fim

13.4 Configurando Regras de Alarme para Métricas Críticas

Esta seção descreve as regras de alarme de algumas métricas e como configurar as regras. Em cenários reais, configure regras de alarme para métricas consultando as seguintes políticas de alarme.

Políticas de alarme para instâncias do DCS Redis

Tabela 13-9 Métricas de instância do DCS Redis para configurar regras de alarme para

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manuseio
Uso da CPU	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Considere a expansão da capacidade com base na análise do serviço. A capacidade da CPU de uma instância de nó único ou principal/em espera não pode ser expandida. Se você precisar de uma capacidade maior, use uma instância de cluster. Essa métrica está disponível somente para instâncias de cluster de proxy, de nó único, principal/em espera. Para instâncias do Cluster Redis, essa métrica está disponível somente no nível do Servidor Redis. Você pode exibir a métrica na página de guia **Redis Server** na página **Performance Monitoring** da instância.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manuseio
Uso médio da CPU	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Considere a expansão da capacidade com base na análise do serviço. A capacidade da CPU de uma instância de nó único ou principal/em espera não pode ser expandida. Se você precisar de uma capacidade maior, use uma instância de cluster. Essa métrica está disponível apenas para instâncias de cluster de proxy, de nó único, principal/em espera e de cluster. Para instâncias do Cluster Redis, essa métrica está disponível somente no nível do Servidor Redis. Você pode exibir a métrica na página de guia **Redis Server** na página **Performance Monitoring** da instância.
Uso da memória	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Crítico	Não	Expanda a capacidade da instância.
Clientes conectados	0–10.000	Limite do alarme: > 8000 Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Otimize o pool de conexões no código de serviço para evitar que o número de conexões exceda o limite máximo. Configure essa política de alarme no nível da instância para instâncias de nó único e principal/em espera. Para instâncias de cluster, configure esta política de alarme no nível do Servidor Redis e Proxy. Para instâncias de nó único e principal/em espera, o número máximo de conexões permitidas é 10.000. Você pode ajustar o limite com base nos requisitos de serviço.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manuseio
Novas Conexões (Contagem/min)	≥ 0	Limite do alarme: > 10.000 Número de períodos consecutivos: 2 Severidade do alarme: menor	-	Verifique se a **connect** é usada e se a conexão do cliente é anormal. Use conexões persistentes ("**pconnect**" na terminologia do Redis) para garantir o desempenho. Configure essa política de alarme no nível da instância para instâncias de nó único e principal/em espera. Para instâncias de cluster, configure esta política de alarme no nível do Servidor Redis e Proxy.
Fluxo de entrada	≥ 0	Limite do alarme: > 80% da largura de banda assegurada Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Sim	Considere a expansão da capacidade com base na análise de serviço e no limite de largura de banda. Configure esse alarme somente para instâncias do DCS Redis 3.0 de nó único e principal/em espera e defina o limite de alarme para 80% da largura de banda assegurada das instâncias do DCS Redis 3.0.
Fluxo de saída	≥ 0	Limite do alarme: > 80% da largura de banda assegurada Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Sim	Considere a expansão da capacidade com base na análise de serviço e no limite de largura de banda. Configure esse alarme somente para instâncias do DCS Redis 3.0 de nó único e principal/em espera e defina o limite de alarme para 80% da largura de banda assegurada das instâncias do DCS Redis 3.0.

Políticas de alarme para instâncias do Memcached DCS

Tabela 13-10 Métricas de instância do Memcached DCS para configurar regras de alarme para

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Uso da CPU	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Verifique o serviço para aumento de tráfego. A capacidade da CPU de uma instância de nó único ou principal/em espera não pode ser expandida. Analise o serviço e considere dividir o serviço ou combinar várias instâncias em um cluster na extremidade do cliente.
Uso da memória	0–100%	Limite do alarme: > 65% Número de períodos consecutivos: 2 Severidade do alarme: menor	Não	Considere expandir a capacidade da instância.
Clientes conectados	0–10.000	Limite do alarme: > 8000 Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Otimize o pool de conexões no código de serviço para evitar que o número de conexões exceda o limite máximo.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Novas conexões	≥ 0	Limite do alarme: > 10.000 Número de períodos consecutivos: 2 Gravidade de alarme: Secundária	-	Verifique se a **connect** é usada e se a conexão do cliente é anormal. Use conexões persistentes ("pconnect" na terminologia do Redis) para garantir o desempenho.
Fluxo de entrada	≥ 0	Limite do alarme: > 80% da largura de banda assegurada Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Sim	Considere a expansão da capacidade com base na análise de serviço e no limite de largura de banda. Para obter detalhes sobre os limites de largura de banda de diferentes especificações de instância, consulte **Especificações da instância do DCS**.
Fluxo de saída	≥ 0	Limite do alarme: > 80% da largura de banda assegurada Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Sim	Considere a expansão da capacidade com base na análise de serviço e no limite de largura de banda. Para obter detalhes sobre os limites de largura de banda de diferentes especificações de instância, consulte **Especificações da instância do DCS**.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Falhas de autenticação	≥ 0	Limite do alarme: > 0 Número de períodos consecutivos: 1 Gravidade de alarme: Crítico	-	Verifique se a senha foi inserida corretamente.

Políticas de alarme para nós do servidor Redis de instâncias do Redis DCS de cluster

Tabela 13-11 Métricas do servidor Redis para configurar políticas de alarme para

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Uso da CPU	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Verifique o serviço para aumento de tráfego. Verifique se o uso da CPU é distribuído uniformemente para os nós do Servidor Redis. Se o uso da CPU for alto em vários nós, considere a expansão da capacidade. Expandir a capacidade de uma instância de cluster dimensionará os nós para compartilhar a pressão da CPU. Se o uso da CPU for alto em um único nó, verifique se as teclas de atalho existem. Se sim, otimize o código de serviço para eliminar teclas de atalho.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Uso médio da CPU	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Considere a expansão da capacidade com base na análise do serviço. A capacidade da CPU de uma instância de nó único ou principal/em espera não pode ser expandida. Se você precisar de uma capacidade maior, use uma instância de cluster. Essa métrica está disponível apenas para instâncias de cluster de proxy, de nó único, principal/em espera e de cluster. Para instâncias do Cluster Redis, essa métrica está disponível somente no nível do Servidor Redis. Você pode exibir a métrica na página de guia **Redis Server** na página **Performance Monitoring** da instância.
Uso da memória	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Verifique o serviço para aumento de tráfego. Verifique se o uso da memória é distribuído uniformemente para os nós do Servidor Redis. Se o uso de memória for alto em vários nós, considere a expansão da capacidade. Se o uso de memória for alto em um único nó, verifique se existem chaves grandes. Em caso afirmativo, otimize o código de serviço para eliminar chaves grandes.
Clientes conectados	0–10.000	Limite do alarme: > 8000 Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Não	Verifique se o número de conexões está dentro do intervalo apropriado. Se sim, ajuste o limiar de alarme.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Novas Conexões	≥ 0	Limite do alarme: > 10.000 Número de períodos consecutivos: 2 Severidade do alarme: menor	-	Verifique se a **connect** é usada. Para garantir o desempenho, use conexões persistentes ("pconnect" na terminologia do Redis).
Logs de consulta lentos	0–1	Limite do alarme: > 0 Número de períodos consecutivos: 1 Gravidade de alarme: Importante	-	Use a função de consulta lenta no console para analisar comandos lentos.

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Uso de largura de banda	0–200%	Limite do alarme: > 90% Número de períodos consecutivos: 2 Gravidade de alarme: Importante	Sim	Verifique se o aumento do uso de largura de banda vem de serviços de leitura ou serviços de gravação com base no fluxo de entrada e saída. Se o uso de largura de banda de um único nó for alto, verifique se existem chaves grandes. Mesmo que o uso da largura de banda exceda 100%, o controle de fluxo pode não necessariamente ser executado. O controle de fluxo real está sujeito à métrica **Flow Control Times**. Mesmo que o uso da largura de banda esteja abaixo de 100%, o controle de fluxo pode ser realizado. O uso da largura de banda em tempo real é relatado uma vez em cada período de relatório. A métrica de tempos de controle de fluxo é relatada a cada segundo. Durante um período de relatório, o tráfego pode aumentar em segundos e, em seguida, voltar a cair. No momento em que o uso da largura de banda é relatado, ele foi restaurado para o nível normal.
Tempos de controle de fluxo	≥ 0	Limite do alarme: > 0 Número de períodos consecutivos: 1 Gravidade de alarme: Crítico	Sim	Considere a expansão da capacidade com base nos limites de especificação, fluxo de entrada e fluxo de saída. **NOTA** Essa métrica é suportada apenas pelo Redis 4.0 e 5.0 e não pelo Redis 3.0.

Políticas de alarme para nós proxy de instâncias de Redis DCS de cluster

Tabela 13-12 Métricas de proxy para configurar políticas de alarme para

Métrica	Intervalo de valores	Política de alarme	Aproximação do limite superior	Sugestão de manipulação
Uso da CPU	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Crítico	Sim	Considere a expansão de capacidade, que adicionará Proxies.
Uso da memória	0–100%	Limite do alarme: > 70% Número de períodos consecutivos: 2 Gravidade de alarme: Crítico	Sim	Considere a expansão de capacidade, que adicionará Proxies.
Clientes conectados	0–30.000	Limite do alarme: > 20.000 Número de períodos consecutivos: 2 Severidade do alarme: Importante	Não	Otimize o pool de conexões no código de serviço para evitar que o número de conexões exceda o limite máximo.

Configurando uma regra de alarme para um grupo de recursos

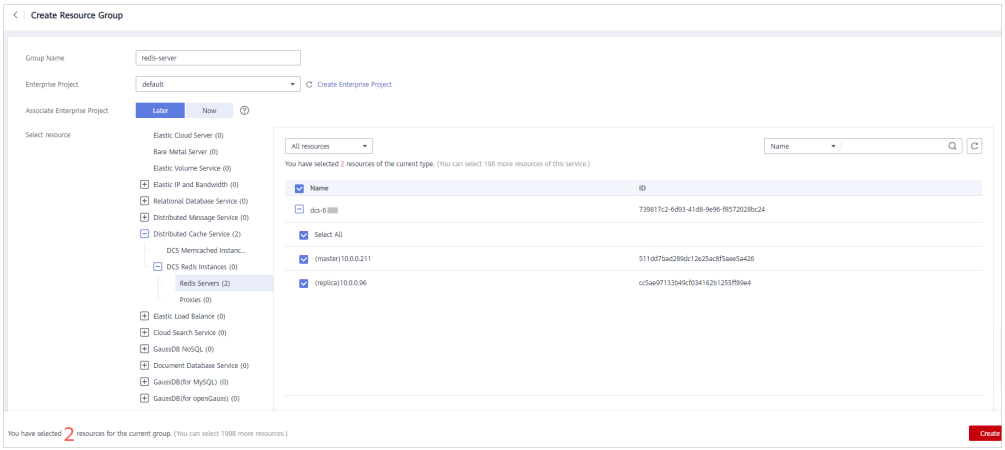
O Cloud Eye permite que você adicione instâncias DCS, nós do servidor Redis e nós proxy a grupos de recursos e gerencie instâncias e regras de alarme por grupo para simplificar o O&M. Para obter detalhes, consulte **Criando um grupo de recursos**.

Passo 1 Criar um grupo de recursos.

1. Faça login no console do Cloud Eye. No painel de navegação, escolha **Resource Groups** and then click **Create Resource Group** no canto superior direito.
2. Insira um nome de grupo e adicione nós do Servidor Redis ao grupo de recursos.

   Você pode adicionar nós do Servidor Redis de instâncias diferentes ao mesmo grupo de recursos.

   Figura 13-1 Criando um grupo de recursos

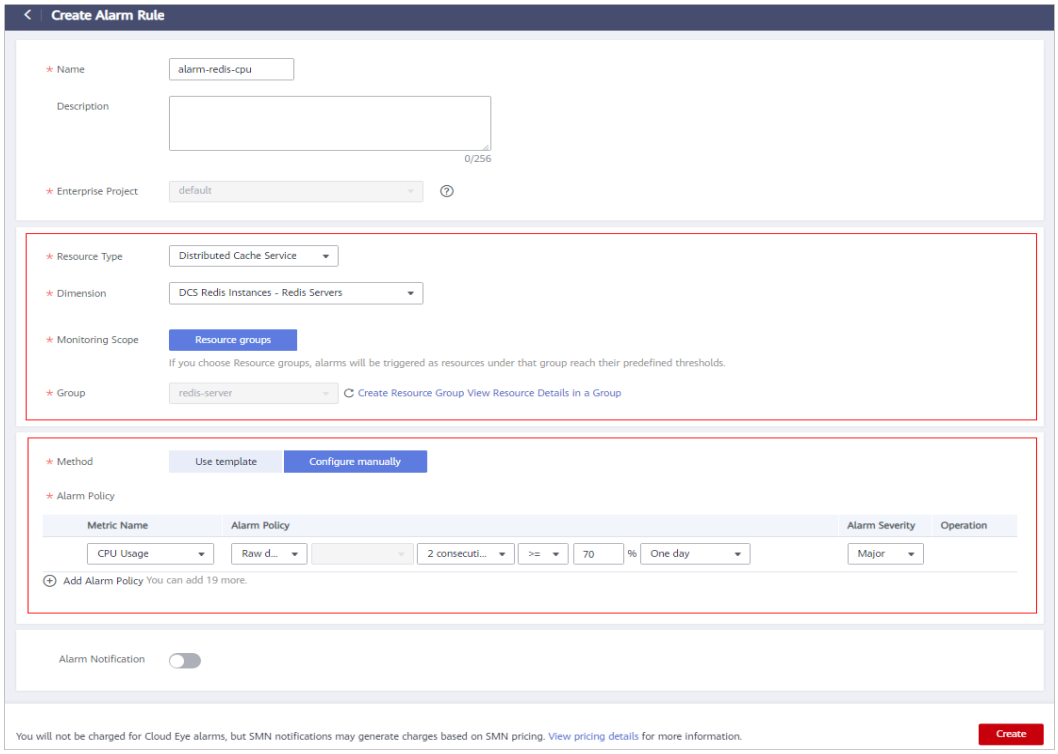


3. Clique em **Create**.

Passo 2 No painel de navegação do console do Cloud Eye, escolha **Alarm Management** > **Alarm Rules** e clique em **Create Alarm Rule** para definir informações de alarme para o grupo de recursos.

Crie uma regra de alarme de uso da CPU para todos os nós do Servidor Redis no grupo de recursos, conforme mostrado na figura a seguir.

Figura 13-2 Criando uma regra de alarme para um grupo de recursos



Passo 3 Clique em **Create**.

----Fim

Configurando uma regra de alarme para um recurso específico

No exemplo a seguir, uma regra de alarme é definida para a métrica **Slow Query Logs** (**is_slow_log_exist**).

Passo 1 Efetue login no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo do console de gerenciamento e selecione uma região.

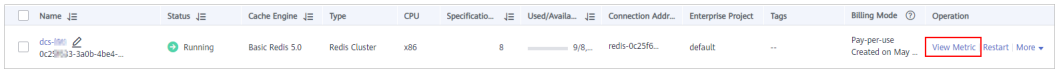
NOTA

Selecione a mesma região que o serviço do aplicativo.

Passo 3 No painel de navegação, escolha **Cache Manager**.

Passo 4 Na linha que contém a instância do DCS cujas métricas você deseja exibir, clique em **View Metric** na coluna **Operation**.

Figura 13-3 Exibição de métricas de instância



Passo 5 Na página exibida, localize a métrica **Slow Query Logs**. Passe o mouse sobre a métrica e clique em  para criar uma regra de alarme para a métrica.

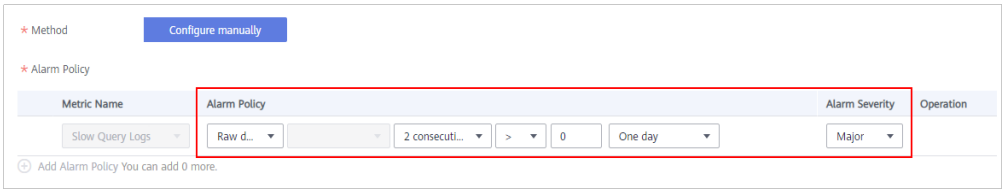
A página **Create Alarm Rule** é exibida.

Passo 6 Especifique as informações do alarme.

1. Defina o nome e a descrição do alarme.
2. Especifique a política de alarme e a severidade do alarme.

   Por exemplo, a política de alarme mostrada em **Figura 13-4** indica que um alarme será disparado se houver consultas lentas na instância por dois períodos consecutivos. Se nenhuma ação for realizada, o alarme será acionado uma vez por dia, até que o valor dessa métrica retorne a **0**.

   Figura 13-4 Configurando o conteúdo do alarme

3. Defina as configurações de notificação de alarme. Se você ativar **Alarm Notification**, defina o período de validade, o objeto de notificação e a condição de gatilho.
4. Clique em **Create**.

NOTA

Para obter mais informações sobre como criar regras de alarme, consulte **Criando uma regra de alarme**.

----Fim

14 Auditoria

14.1 Operações registradas pelo CTS

Com o CTS, você pode consultar, auditar e revisar as operações realizadas nos recursos da nuvem. Os rastreamentos incluem as solicitações de operação enviadas usando o console de gerenciamento ou as API abertas, bem como os resultados dessas solicitações.

A seguir, são listadas as operações DCS que podem ser registradas pelo CTS.

Tabela 14-1 Operações DCS que podem ser gravadas pelo CTS

Operação	Tipo de recurso	Nome do Rastreamento
Criando uma instância	Redis	createDCSInstance
Enviando uma solicitação de criação de instância	Redis	submitCreateDCSInstanceRequest
Excluindo várias instâncias	Redis	batchDeleteDCSInstance
Deletando uma Instância	Redis	deleteDCSInstance
Modificando informações da instância	Redis	modifyDCSInstanceInfo
Modificando configurações de instância	Redis	modifyDCSInstanceConfig

Operação	Tipo de recurso	Nome do Rastreamento
Alterando a senha da instância	Redis	modifyDCSInstancePassword
Interrompendo uma instância	Redis	stopDCSInstance
Submetendo uma solicitação de interrupção de instância	Redis	submitStopDCSInstanceRequest
Reiniciando uma Instância	Redis	restartDCSInstance
Enviando uma instância reiniciando a solicitação	Redis	submitRestartDCSInstanceRequest
Iniciando uma instância	Redis	startDCSInstance
Enviando uma solicitação inicial de instância	Redis	submitStartDCSInstanceRequest
Limpando dados da instância	Redis	flushDCSInstance
Interrompendo várias instâncias	Redis	batchStopDCSInstance
Submetendo uma solicitação para interromper instâncias em lotes	Redis	submitBatchStopDCSInstanceRequest
Reiniciando instâncias em lotes	Redis	batchRestartDCSInstance

Operação	Tipo de recurso	Nome do Rastreamento
Enviando uma solicitação para reiniciar instâncias em lotes	Redis	submitBatchRestartDCSInstanceRequest
Iniciando várias instâncias	Redis	batchStartDCSInstance
Enviando uma solicitação para iniciar instâncias em lotes	Redis	submitBatchStartDCSInstanceRequest
Restaurando dados da instância	Redis	restoreDCSInstance
Enviando uma solicitação para restaurar dados da instância	Redis	submitRestoreDCSInstanceRequest
Fazendo backup de dados da instância	Redis	backupDCSInstance
Enviando uma solicitação para fazer backup de dados da instância	Redis	submitBackupDCSInstanceRequest
Exclusão de arquivos de backup da instância	Redis	deleteInstanceBackupFile
Exclusão de tarefas em segundo plano	Redis	deleteDCSInstanceJobRecord
Modificando especificações de instância	Redis	modifySpecification

Operação	Tipo de recurso	Nome do Rastreamento
Enviando uma solicitação para modificar especificações de instância	Redis	submitModifySpecificationRequest
Criando um pedido de assinatura de instância	Redis	createInstanceOrder
Criando um pedido para modificar especificações de instância	Redis	createSpecificationChangeOrder
Atualizando a ID do projeto empresarial	Redis	updateEnterpriseProjectId
Alternando entre os nós mestre e stand-by	Redis	masterStandbySwitchover
Desativação do acesso público	Redis	disablePublicNetworkAccess
Ativando o acesso público	Redis	enablePublicNetworkAccess
Redefinindo a senha da instância	Redis	resetDCSInstancePassword
Enviando uma solicitação para limpar dados de instância	Redis	submitFlushDCSInstanceRequest
Acessando a CLI da Web	Redis	webCliLogin
Executando comandos na CLI da Web	Redis	webCliCommand
Saindo da CLI da Web	Redis	webCliLogout

Operação	Tipo de recurso	Nome do Rastreamento
Migração de dados offline	Redis	offlineMigrate
Alterando o modo de cobrança	Redis	billingModeChange

Para obter detalhes sobre como exibir logs de auditoria, consulte **Consultando rastreamentos em tempo real**.

