Distributed Cache Service

Perguntas frequentes

Edição 01

Data 2023-12-20

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.

Huawei Cloud Computing Technologies Co., Ltd.

Endereço: Huawei Cloud Data Center, Rua Jiaoxinggong
Avenida Qianzhong
Novo Distrito de Gui'an
Guizhou 550029
República Popular da China

Site: https://www.huaweicloud.com/intl/pt-br/

Índice

1 Tipos/versões de instâncias

1.1 Comparação entre o Redis e o Memcached

O Redis e o Memcached são bancos de dados em memória de código aberto populares, fáceis de usar e com desempenho superior ao dos bancos de dados relacionais.

Como selecionar entre os dois bancos de dados de chave-valor?

O Memcached é adequado para armazenar estruturas de dados simples, enquanto o Redis é adequado para armazenar dados mais complexos e maiores que exigem persistência.

Para obter detalhes, consulte a tabela a seguir.

Tabela 1-1 Diferenças entre Redis e Memcached

Item	Redis	Memcached
Latência	Banco de dados em memória com latência de sub-milissegundos	Banco de dados em memória com latência de sub-milissegundos
Fácil de usar	Sintaxe simples e fácil de usar	Sintaxe simples e fácil de usar
Armazenamento distribuído	Expansão horizontal no modo de cluster	Suportado
Cliente multilíngue	Suporta conexões de clientes em mais de 30 linguagens, incluindo Java, C e Python.	Suporta conexões de clientes em mais de 10 linguagens, incluindo Java, C e Python.
Tópico/ processo	Núcleo único e thread único Comunicação de thread único, evitando alternância e contenção de contexto desnecessárias I/O sem bloqueio (multiplexação de I/O) é usada para reduzir o consumo de recursos quando vários clientes estão conectados.	Multi-thread e dimensionável O desempenho do Memcached pode ser melhorado aumentando o número de CPUs. Há uma vantagem de desempenho óbvia no cenário em que o valor da chave é grande.

Item	Redis	Memcached
Armazenamento persistente	Suportado Cada operação de gravação (adicionar, excluir ou modificar dados) pode ser gravada em disco (arquivo AOF).	Suportado **NOTA** A persistência não é suportada pelo Memcached de código aberto, mas é suportada pelo DCS for Memcached da HUAWEI CLOUD.
Estrutura de dados	Suporta estruturas de dados complexas, como hash, lista, conjunto e conjunto classificado, atendendo a vários cenários.	Suporta cadeias simples.
Suporte a script Lua	Suportado	Não suportado
Backup de snapshot	Suportado Snapshots são gerados periodicamente. Portanto, não há garantia de que os dados não serão perdidos. O Redis bifurca um subprocesso para gerar instantâneos. Quando há uma grande quantidade de dados, o serviço de Redis pode ser interrompido por um curto período de tempo.	Não suportado
Migração de dados	Suportado Os dados podem ser copiados e migrados para uma nova instância do Redis por meio da restauração de snapshot RDB ou da reprodução de arquivos AOF.	Não suportado
Restrição do valor da chave	O valor de uma chave pode ser de até 1 GB.	1 MB
Vários bancos de dados	Uma instância do DCS Redis de nó único ou principal/em espera suporta até 256 bancos de dados do Redis. Uma instância de Proxy Cluster ou Redis Cluster suporta apenas uma base de dados, ou seja, DB0.	Não suportado

Com base na comparação anterior, tanto o Redis quanto o Memcached são fáceis de usar e têm alto desempenho. No entanto, o Redis e o Memcached são diferentes em termos de armazenamento de estrutura de dados, persistência, backup, migração e suporte a scripts. É

aconselhável selecionar o mecanismo de cache mais apropriado com base em cenários reais de aplicações.

NOTA

O Memcached é adequado para cenários de cache de pequena quantidade de dados estáticos, onde os dados são lidos apenas sem computação e processamento adicionais, por exemplo, trechos de código HTML.

O Redis possui estruturas de dados mais ricas e cenários de aplicações mais amplos.

1.2 Comparação de versões

Ao criar uma instância do DCS Redis, você pode selecionar a versão do mecanismo de cache e o tipo de instância.

NOTA

O DCS for Redis 3.0 não é mais fornecido. Você pode usar o DCS for Redis 4.0, 5.0 ou 6.0.

- **Versão**

  O DCS suporta Redis 6.0, 5.0, 4.0 e 3.0. **Tabela 1-2** descreve as diferenças entre essas versões. Para obter detalhes sobre os novos recursos do Redis 4.0 e 5.0, consulte **Novos recursos do DCS for Redis 4.0** and **New Features of DCS for Redis 5.0** .

Tabela 1-2 Diferenças entre as versões do Redis

Características	Redis 3.0	Redis 4.0 e Redis 5.0	Redis 6.0
Compatibilidade de código aberto	Redis 3.0.7	Redis 4.0.14 e 5.0.14, respectivamente **NOTA** As instâncias do DCS Redis 5.0 criadas antes de fevereiro de 2022 são compatíveis com o Redis 5.0.9 de código aberto. ● Para obter detalhes sobre como consultar a versão de código aberto, consulte **Como exibir a versão de uma instância do DCS Redis?** ● Para usar o Redis 5.0.14, crie outra instância. Atualmente, a versão do Redis não pode ser atualizada.	Edição básica: Redis 6.2.7 Edição profissional: KeyDB 6.0.16
Modo de implementação da instância	Baseado em VMs	Containerizado com base em servidores físicos	Containerizado com base em servidores físicos
Arquitetura da CPU	x86	x86	x86

Características	Redis 3.0	Redis 4.0 e Redis 5.0	Redis 6.0
Tempo necessário para criar uma instância	3 a 15 minutos ou 10 a 30 minutos para instâncias de cluster.	8 segundos	8 segundos
QPS	100.000 QPS por nó	100.000 QPS por nó	Edição básica: 150.000 QPS por nó Edição profissional: 400.000 QPS por nó
Acesso à rede pública	Suportado	Não suportado	Não suportado
Conexão de nome de domínio	Suportado em VPC	Suportado em VPC	Suportado em VPC
Gerenciamento de dados visualizados	Não suportado	CLI da Web para acesso ao Redis e gerenciamento de dados.	CLI da Web para acesso ao Redis e gerenciamento de dados.
Tipos de instância	Nó único, principal/em espera e Proxy Cluster	Nó único, principal/em espera, Proxy Cluster e Redis Cluster	Principal/em espera apenas
Memória total da instância	Varia de 2 GB a 1024 GB.	As especificações regulares variam de 2 GB a 1024 GB. Especificações pequenas de 128 MB, 256 MB, 512 MB e 1 GB também estão disponíveis para instâncias de nó único e principal/em espera.	4 GB, 8 GB, 16 GB, 32 GB e 64 GB (128 MB, 256 MB, 512 MB e 1 GB são compatíveis adicionalmente com instâncias de nó único e principal/em espera)
Expansão/redução de capacidade	Expansão e redução da capacidade on-line	Expansão e redução da capacidade on-line	Expansão e redução da capacidade on-line

Características	Redis 3.0	Redis 4.0 e Redis 5.0	Redis 6.0
Backup e restauração	Suportado para instâncias de nó principal/em espera e de Proxy Cluster	Suportado para instâncias principal/em espera, Proxy Cluster e Redis Cluster	Suportado para instâncias de nó principal/em espera

NOTA

As arquiteturas subjacentes variam de acordo com a versão do Redis. Depois que uma versão do Redis é escolhida, ela não pode ser alterada. Por exemplo, você não pode atualizar uma instância de DCS Redis 3.0 para Redis 4.0 ou 5.0. Se você precisar de uma versão superior do Redis, compre uma nova instância que atenda aos seus requisitos e, em seguida, migre os dados da instância antiga para a nova.

- **Tipos de instância**

  DCS fornece tipos de instância de nó único, principal/em espera, Proxy Cluster e Redis Cluster. Para obter detalhes sobre suas arquiteturas e cenários de aplicações, consulte **Tipos de instância de DCS**.

1.3 Novos recursos do DCS for Redis 4.0

Em comparação com o DCS for Redis 3.0, o DCS for Redis 4.0 e versões posteriores adicionam suporte aos novos recursos do Redis de código aberto e suportam a criação mais rápida de instâncias.

A implementação da instância mudou do modo de VM para o modo de conteinerização baseado em servidor físico. Uma instância pode ser criada dentro de 8 a 10 segundos.

O Redis 4.0 oferece os seguintes novos recursos:

1. Novos comandos, como **MEMORY** e **SWAPDB**
2. Lazyfree, atrasando a exclusão de chaves grandes e reduzindo o impacto da exclusão nos recursos do sistema
3. Otimização do desempenho da memória, ou seja, desfragmentação ativa

Comando MEMORY

No Redis 3.0 e versões anteriores, você pode executar o comando **INFO MEMORY** para aprender apenas as estatísticas de memória limitada. O Redis 4.0 introduz o comando **MEMORY** para ajudá-lo a entender melhor o uso da memória do Redis.

```
127.0.0.1:6379[8]> memory help
1) MEMORY <subcommand> arg arg ... arg. Subcommands are:
2) DOCTOR - Return memory problems reports.
3) MALLOC-STATS -- Return internal statistics report from the memory allocator.
4) PURGE -- Attempt to purge dirty pages for reclamation by the allocator.
5) STATS -- Return information about the memory usage of the server.
6) USAGE <key> [SAMPLES <count>] -- Return memory in bytes used by <key> and its
value. Nested values are sampled up to <count
> times (default: 5).
127.0.0.1:6379[8]>
```


usage

Digite **memory usage *[key]***. Se a chave existir, a memória estimada usada pelo valor da chave é retornada. Se a chave não existir, **nil** é retornado.

```
127.0.0.1:6379[8]> set dcs "DCS is an online, distributed, in-memory cache
service compatible with Redis, and Memcached."
OK
127.0.0.1:6379[8]> memory usage dcs
(integer) 141
127.0.0.1:6379[8]>
```

NOTA

1. O **usage** coleta estatísticas sobre o uso de memória do valor e da chave, excluindo o uso de memória Expire da chave.

   ```
   // The following is verified based on Redis 5.0.2. Results may differ in
   other Redis versions.
   192.168.0.66:6379> set a "Hello, world!"
   OK
   192.168.0.66:6379> memory usage a
   (integer) 58
   192.168.0.66:6379> set abc "Hello, world!"
   OK
   192.168.0.66:6379> memory usage abc
   (integer) 60   //After the key name length changes, the memory usage also
   changes. This indicates that the usage statistics contain the usage of the
   key.
   192.168.0.66:6379> expire abc 1000000
   (integer) 1
   192.168.0.66:6379> memory usage abc
   (integer) 60   // After the expiration time is added, the memory usage
   remains unchanged. This indicates that the usage statistics do not contain
   the expire memory usage.
   192.168.0.66:6379>
   ```

2. Para hashes, lists, sets e sorted sets, o comando **MEMORY USAGE** mostra estatísticas e fornece o uso estimado da memória.

   Uso: **memory usage** *keyset* **samples** *1000*

   keyset indica a chave de um conjunto, e *1000* indica o número de amostras.

stats

Retorna o uso detalhado de memória da instância atual.

Uso: **memory stats**

```
127.0.0.1:6379[8]> memory stats
 1) "peak.allocated"
 2) (integer) 2412408
 3) "total.allocated"
 4) (integer) 2084720
 5) "startup.allocated"
 6) (integer) 824928
 7) "replication.backlog"
 ... ...
```

A tabela a seguir descreve os significados de alguns itens de retorno.

Tabela 1-3 Valores de retorno de MEMORY STATS

Valor retornado	Descrição
peak.allocated	Memória de pico alocada pelo alocador durante a execução da instância do Redis. É o mesmo que **used_memory_peak** da **info memory**.
total.allocated	O número de bytes alocados pelo alocador. É o mesmo que **used_memory** of **info memory**
startup.allocated	Quantidade inicial de memória consumida pelo Redis na inicialização em bytes
replication.backlog	Tamanho em bytes da lista de pendências de replicação. É especificado no parâmetro **repl-backlog-size**. O valor padrão é **1 MB**.
clients.slaves	O tamanho total em bytes de todas as réplicas de custos indiretos
clients.normal	O tamanho total em bytes de todos os clientes de custos indiretos
overhead.total	A soma de todos os custos indiretos. **overhead.total** é o total de memória **total.allocated** alocado pelo alocador menos a memória real usada para armazenar dados.
keys.count	O número total de chaves armazenadas em todos os bancos de dados no servidor
keys.bytes-per-key	Número médio de bytes ocupados por cada chave. Observe que o custo indireto também é alocado para cada chave. Portanto, esse valor não indica o comprimento médio da chave.
dataset.bytes	Bytes de memória ocupados por dados do Redis, ou seja, **overhead.total** subtraído de **total.allocated**
dataset.percentage	A porcentagem de **dataset.bytes** fora do uso de memória líquida
peak.percentage	A porcentagem de pico.alocado fora do **total.allocated**
fragmentation	Taxa de fragmentação de memória

doctor

Uso: **memory doctor**

Se o valor de **used_memory** (**total.allocated**) for menor que 5 MB, **MEMORY DOCTOR** considera que o uso de memória é muito pequeno e não realiza diagnósticos adicionais. Se qualquer uma das seguintes condições for atendida, o Redis fornecerá resultados de diagnóstico e sugestões:

1. O pico de memória alocada é maior que 1,5 vezes do **total_allocated** atual, ou seja, **peak.allocated**/**total.allocated** > 1,5, indicando que a taxa de fragmentação da memória é alta e que o RSS é muito maior que **used_memory**.
2. O valor de alta fragmentação/fragmentação é maior que 1,4, indicando que a taxa de fragmentação da memória é alta.

3. O uso médio de memória de cada cliente normal é superior a 200 KB, indicando que o pipeline pode ser usado incorretamente ou que o cliente Pub/Sub não processa mensagens a tempo.
4. O uso médio de memória de cada cliente secundário é maior que 10 MB, indicando que o tráfego de gravação do principal é muito alto.

purge

Uso: **memory purge**

Executa o comando interno **jemalloc** para liberar a memória. Os objetos liberados incluem a memória que é ocupada, mas não usada pelos processos do Redis, ou seja, fragmentos de memória.

NOTA

MEMORY PURGE aplica-se somente à instância do Redis que usa **jemalloc** como alocador.

Lazyfree

Problema

O Redis é de thread único. Quando uma solicitação demorada é executada, todas as solicitações são colocadas na fila. Antes que a solicitação seja concluída, o Redis não pode responder a outras solicitações. Como resultado, podem ocorrer problemas de desempenho. Uma das solicitações demoradas é a exclusão de uma chave grande.

Princípio

O recurso Lazyfree do Redis 4.0 evita o bloqueio causado pela exclusão de chaves grandes, garantindo desempenho e disponibilidade.

Ao excluir uma chave, o Redis libera de forma assíncrona a memória ocupada pela chave. A operação de liberação da chave é processada no subthread de I/O de segundo plano (BIO).

Utilização

1. Exclusão ativa
   - **UNLINK**

     Semelhante ao **DEL**, este comando remove chaves. Se houver mais de 64 elementos a serem excluídos, a operação de liberação de memória é executada em um thread de BIO independente. Portanto, o comando **UNLINK** pode excluir uma chave grande contendo milhões de elementos em um curto espaço de tempo.
   - **FLUSHALL** e **FLUSHDB**

     Uma opção **ASYNC** foi adicionada ao **FLUSHALL** e ao **FLUSHDB** para permitir que todo o conjunto de dados ou um único banco de dados fosse liberado de forma assíncrona.
2. Exclusão passiva: exclusão de chaves expiradas e remoção de chaves grandes

   Há quatro cenários para exclusão passiva e cada cenário corresponde a um parâmetro. Esses parâmetros são desativados por padrão.

   ```
   lazyfree-lazy-eviction no // Whether to enable Lazyfree when the Redis memory
   usage reaches maxmemory and the eviction policy is set.
   lazyfree-lazy-expire no // Whether to enable Lazyfree when the key with TTL
   is going to expire.
   lazyfree-lazy-server-del no // An implicit DEL key is used when an existing
   ```



```
key is processed.
slave-lazy-flush no // Perform full data synchronization for the standby
node. Before loading the RDB file of the master, the standby node executes
the FLUSHALL command to clear its own data.
```

Outros novos comandos

1. **swapdb**

   Troca dois bancos de dados de Redis.

   swapdb *dbindex1 dbindex2*

2. **zlexcount**

   Retorna o número de elementos no conjunto classificado.

   zlexcount *key min max*

Otimização de memória e desempenho

1. Em comparação com antes, a mesma quantidade de dados pode ser armazenada com menos memória.
2. A memória usada pode ser desfragmentada e gradualmente removida.

1.4 Novos recursos do DCS for Redis 5.0

O DCS for Redis 5.0 é compatível com os novos recursos do Redis 5.0 de código aberto, além de todas as melhorias e novos comandos no Redis 4.0.

Estrutura de dados do Stream

Stream é um novo tipo de dados introduzido com o Redis 5.0. Suporta persistência de mensagens e multicast.

Figura 1-1 mostra a estrutura de um Stream do Redis, que permite que as mensagens sejam anexadas ao fluxo.

Streams têm os seguintes recursos:

1. Um Stream pode ter vários grupos de consumidores.
2. Cada grupo de consumidores contém um **Last_delivered_id** que aponta para o último item consumido (mensagem) no grupo de consumidores.
3. Cada grupo de consumidores contém vários consumidores. Todos os consumidores compartilham o **last_delivered_id** do grupo de consumidores. Uma mensagem pode ser consumida por apenas um consumidor.
4. **pending_ids** no consumidor pode ser usado para registrar os IDs de itens que foram enviados ao cliente, mas não foram confirmados.
5. Para uma comparação detalhada entre o Stream e outras estruturas de dados do Redis, consulte **Tabela 1-4**.

Figura 1-1 Estrutura de dados de Stream

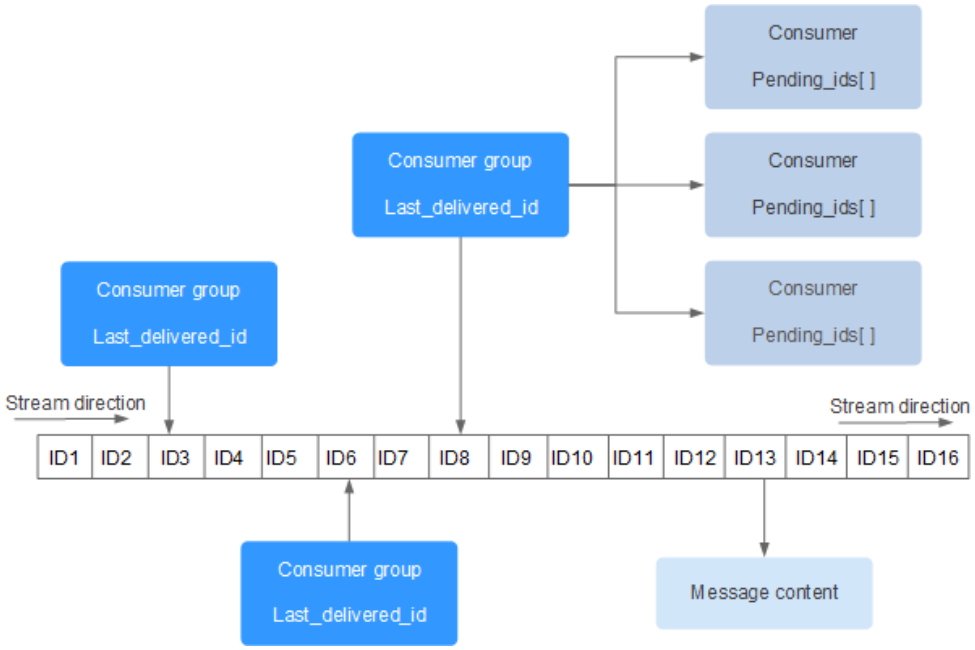


Tabela 1-4 Diferenças entre Streams e estruturas de dados de Redis existentes

Item	Stream	List, Pub/Sub, Zset
Complexidade da busca de itens	O(log(N))	List: O(N)
Offset	Compatível. Cada item tem um ID exclusivo. O ID não é alterado à medida que outros itens são adicionados ou despejados.	List: incompatível. Se um item for despejado, o item mais recente não poderá ser localizado.
Persistência	Compatível. Streams são persistidos para arquivos AOF e RDB.	Pub/Sub: incompatível.
Grupo de consumidor	Compatível.	Pub/Sub: incompatível.
Reconhecimento	Compatível.	Pub/Sub: incompatível.
Desempenho	Não tem relação com o número de consumidores.	Pub/Sub: positivamente relacionado ao número de clientes.
Despejo	Streams são eficientes em memória, bloqueando para despejar os dados que são muito antigos e usando uma árvore radix e listpack.	O Zset consome mais memória porque não suporta inserir os mesmos itens, bloquear ou despejar dados

Item	Stream	List, Pub/Sub, Zset
Excluir itens aleatoriamente	Incompatível.	Zset: compatível.

Comandos de Stream

Os comandos de Stream são descritos abaixo na ordem em que são usados. Para mais detalhes, consulte **Tabela 1-5**.

1. Execute o comando **XADD** para adicionar um item de fluxo, ou seja, criar um Stream. O número máximo de mensagens que podem ser salvas pode ser especificado ao adicionar o item.
2. Crie um grupo de consumidores executando o comando **XGROUP**.
3. Um consumidor usa o comando **XREADGROUP** para consumir mensagens.
4. Após o consumo, o cliente executa o comando **XACK** para confirmar que o consumo é bem sucedido.

Figura 1-2 Comandos de Stream

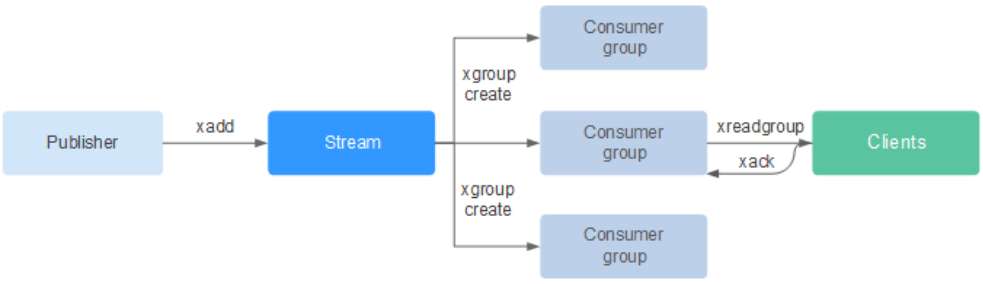


Tabela 1-5 Descrição dos comandos de Stream

Comando	Descrição	Sintaxe
XACK	Exclui uma ou várias mensagens da *pending entry list* (PEL) um grupo de consumidores do fluxo.	XACK key group ID [ID ...]
XADD	Adiciona uma entrada especificada ao fluxo em uma chave especificada. Se a chave não existir, executar este comando resultará em uma chave a ser criada automaticamente com base na entrada.	XADD key ID field string [field string ...]
XCLAIM	Altera a propriedade de uma mensagem pendente, para que o novo proprietário seja o consumidor especificado como argumento do comando.	XCLAIM key group consumer min-idle-time ID [ID ...] [IDLE ms] [TIME ms-unix-time] [RETRYCOUNT count] [FORCE] [JUSTID]

Comando	Descrição	Sintaxe
XDEL	Remove as entradas especificadas de um fluxo e retorna o número de entradas excluídas, que pode ser diferente do número de IDs passados ao comando no caso de determinados IDs não existirem.	XDEL key ID [ID ...]
XGROUP	Gerencia os grupos de consumidores associados a um stream. Você pode usar **XGROUP** para: ● Criar um novo grupo de consumidores associado a um fluxo. ● Destruir um grupo de consumidores. ● Remover um consumidor especificado de um grupo de consumidores. ● Definir o *ID da última entrega* do grupo de consumidores para outra coisa.	XGROUP [CREATE key groupname id-or-$] [SETID key id-or-$] [DESTROY key groupname] [DELCONSUMER key groupname consumername]
XINFO	Recupera informações diferentes sobre os fluxos e grupos de consumidores associados.	XINFO [CONSUMERS key groupname] key key [HELP]
XLEN	Retorna o número de entradas em um fluxo. Se a chave especificada não existir, **0** é retornado, indicando um fluxo vazio.	XLEN key
XPENDING	Obtém dados de um fluxo através de um grupo de consumidores. Esse comando é a interface para inspecionar a lista de mensagens pendentes para observar e entender quais clientes estão ativos, quais mensagens estão pendentes para serem consumidas ou para ver se há mensagens ociosas.	XPENDING key group [start end count] [consumer]
XRANGE	Retorna entradas correspondentes a um determinado intervalo de IDs.	XRANGE key start end [COUNT count]
XREAD	Lê dados de um ou vários fluxos, retornando apenas entradas com um ID maior que o último ID recebido informado pelo chamador.	XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]

Comando	Descrição	Sintaxe
XREADGROUP	Uma versão especial do comando **XREAD**, que é usada para especificar um grupo de consumidores para ler.	XREADGROUP GROUP group consumer [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]
XREVRANGE	Este comando é exatamente como **XRANGE**, mas com a diferença notável de retornar as entradas na ordem inversa e também tomar o intervalo de início-fim na ordem inversa.	XREVRANGE key end start [COUNT count]
XTRIM	Apara o fluxo para um número especificado de itens, se necessário, despejando itens antigos (itens com IDs inferiores).	XTRIM key MAXLEN [~] count

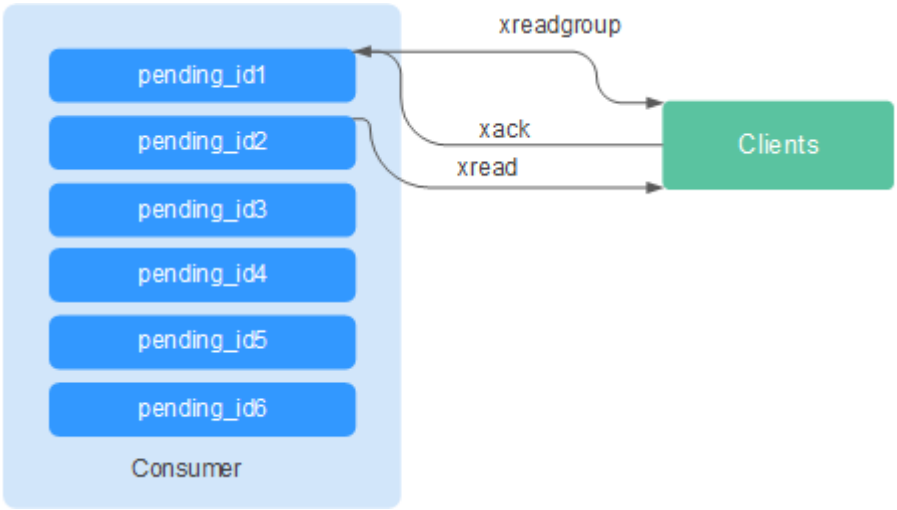
Confirmação de mensagem (item de fluxo)

Em comparação com o Pub/Sub, Streams não apenas suportam grupos de consumidores, mas também o reconhecimento de mensagens.

Quando um consumidor invoca o comando **XREADGROUP** para ler ou invoca o comando **XCLAIM** para assumir uma mensagem, o servidor não sabe se a mensagem é processada pelo menos uma vez. Portanto, depois de ter processado com sucesso uma mensagem, o consumidor deve invocar o comando **XACK** para notificar o Stream de modo que a mensagem não será processada novamente. Além disso, a mensagem é removida do PEL e a memória será liberada do servidor Redis.

Em alguns casos, como falhas de rede, o cliente não invoca **XACK** após o consumo. Nesses casos, o ID do item é mantido no PEL. Depois que o cliente é reconectado, defina o ID da mensagem inicial de **XREADGROUP** para 0-0, indicando que todas as mensagens PEL e mensagens após **last_id** são lidas. Além disso, a transmissão repetida de mensagens deve ser suportada quando os consumidores consomem mensagens.

Figura 1-3 Mecanismo de reconhecimento



Otimização do uso da memória

O uso de memória do Redis 5.0 é otimizado com base na versão anterior.

- Desfragmentação ativa

  Se uma chave for modificada com frequência e o comprimento do valor mudar constantemente, o Redis alocará memória adicional para a chave. Para obter alto desempenho, o Redis usa o alocador de memória para gerenciar a memória. A memória nem sempre é liberada para o sistema operacional. Como resultado, ocorrem fragmentos de memória. Se a taxa de fragmentação (**used_memory_rss**/**used_memory**) for maior que 1,5, o uso da memória é ineficiente.

  Para reduzir fragmentos de memória, planeje e use adequadamente os dados de cache e padronize a gravação de dados.

  Para o Redis 3.0 e versões anteriores, os problemas de fragmentação de memória são resolvidos reiniciando o processo regularmente. Recomenda-se que os dados de cache reais não excedam 50% da memória disponível.

  Para o Redis 4.0, a desfragmentação ativa é suportada e a memória é desfragmentada enquanto estiver on-line. Além disso, o Redis 4.0 oferece suporte à desfragmentação manual de memória executando o comando **memory purge**.

  Para o Redis 5.0, a desfragmentação ativa aprimorada é compatível com o Jemalloc atualizado, que é mais rápido, mais inteligente e oferece menor latência.

- Melhorias na implementação do HyperLogLog

  Um HyperLogLog é uma estrutura de dados probabilística usada para calcular a cardinalidade de um conjunto enquanto consome pouca memória. O Redis 5.0 melhora o HyperLogLog otimizando ainda mais o uso da memória.

  Por exemplo: a árvore B é eficiente na contagem, mas consome muita memória. Ao usar o HyperLogLog, é possível economizar muita memória. Enquanto a árvore B requer 1 MB de memória para contagem, o HyperLogLog precisa de apenas 1 KB.

- Estatísticas de memória aprimoradas

  A informação retornada pelo comando **INFO** é mais detalhada.

Novos e melhores comandos

1. **Gerenciamento aprimorado de clientes**
   - O redis-cli suporta o gerenciamento de cluster.

     No Redis 4.0 e versões anteriores, o módulo **redis-trib** precisa ser instalado para gerenciar clusters.

     O Redis 5.0 otimiza o redis-cli, integrando todas as funções de gerenciamento de cluster. Você pode executar o comando **redis-cli --cluster help** para obter mais informações.
   - O desempenho do cliente é aprimorado em cenários frequentes de conexão e desconexão.

     Essa otimização é valiosa quando sua aplicação precisa usar conexões curtas.
2. **Uso mais simples de conjuntos ordenados**

   Os comandos **ZPOPMIN** e **ZPOPMAX** são adicionados para os conjuntos ordenados.
   - ZPOPMIN key [count]

     Remove e retorna até **count** os membros com as pontuações mais baixas no conjunto classificado armazenado na **key**. Ao retornar vários elementos, aquele com

a pontuação mais baixa será o primeiro, seguido pelos elementos com pontuações mais altas.

– ZPOPMAX key [count]

Remove e retorna até **count** os membros com as pontuações mais altas no conjunto classificado armazenado na **key**. Ao retornar vários elementos, aquele com a pontuação mais baixa será o primeiro, seguido pelos elementos com pontuações mais baixas.

3. **Mais subcomandos adicionados ao comando help**

   O comando **help** pode ser usado para visualizar informações de ajuda, poupando-lhe o trabalho de visitar **redis.io** todas as vezes. Por exemplo, execute o seguinte comando para exibir as informações de ajuda do stream: **xinfo help**

```
127.0.0.1:6379> xinfo help
1) XINFO <subcommand> arg arg ... arg. Subcommands are:
2) CONSUMERS <key> <groupname>  -- Show consumer groups of group <groupname>.
3) GROUPS <key>                 -- Show the stream consumer groups.
4) STREAM <key>                 -- Show information about the stream.
5) HELP                         -- Print this help.
127.0.0.1:6379>
```

4. **Dicas de entrada de comandos redis-cli**

   Depois de inserir um comando completo, o redis-cli exibe uma dica de parâmetro para ajudá-lo a memorizar o formato de sintaxe do comando.

   Como mostrado na figura a seguir, execute o comando **zadd** e o redis-cli exibe a sintaxe **zadd** na cor clara.

RDB que armazena informações de LFU e LRU

No Redis 5.0, as políticas de remoção de chaves de armazenamento **LRU** e **LFU** foram adicionadas ao arquivo de snapshot do RDB.

- FIFO: primeiro a entrar, primeiro a sair. Os primeiros dados armazenados são despejados primeiro.
- LRU: menos usado recentemente. Os dados que não são usados há muito tempo são despejados primeiro.
- LFU: menos frequentemente usado. Os dados usados com menos frequência são despejados primeiro.

NOTA

O formato de arquivo RDB do Redis 5.0 foi modificado e é compatível com versões anteriores. Portanto, se um snapshot for usado para migração, os dados poderão ser migrados das versões anteriores do Redis para o Redis 5.0, mas não poderão ser migrados do Redis 5.0 para as versões anteriores.

1.5 Quais são as diferenças entre o DCS for Redis baseado em ARM e baseado em x86?

O DCS for Redis suporta totalmente as arquiteturas de CPU ARM e x86. Eles não diferem em funções ou compatibilidade com o cliente.

No entanto, Kunpeng e x86 diferem nos seguintes aspectos:

- Versões do Redis suportadas
  - Redis baseado em Arm: Redis 4.0 e Redis 5.0
  - Redis baseado em x86: Redis 6.0, 5.0, 4.0 e 3.0
- Tipos de instância suportados
  - Arm: nó único, principal/em espera e Redis Cluster
  - x86: Proxy Cluster de nó único, principal/em espera, Redis 3.0 e Redis Cluster 4.0 ou 5.0
- Preços

  O Redis baseado em Kunpeng é 30% mais barato que o Redis baseado em x86.
- Desempenho

  O desempenho de diferentes especificações de instância está listado em **Especificações da instância de DCS**.

O Redis baseado em x86 oferece maior desempenho de CPU única do que o Redis baseado em ARM em cenários que envolvem comandos complexos, como chaves grandes ou chaves cuja complexidade de tempo é maior que O(N).

Em conclusão, tanto o Redis baseado em ARM quanto o Redis baseado em x86 oferecem desempenho capaz de atender às suas necessidades de serviço, mas o Redis baseado em ARM é mais econômico.

1.6 Posso mudar a arquitetura da CPU?

Não.

Se você quiser usar uma arquitetura de CPU diferente, crie outra instância com a arquitetura de CPU desejada e migre os dados.

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

- A instância de destino está disponível. Se você já tiver uma instância do DCS Redis, use-a diretamente e limpe os dados da instância antes da migração. Para obter detalhes, consulte **Limpeza de dados de instância de DCS**.

Se os dados da instância de destino não forem limpos 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.

- Os recursos do Redis de destino, do Redis de origem e da tarefa de migração estão na mesma VPC.

  NOTA

  Se as instâncias do Redis de destino e de origem não estiverem na mesma VPC, verifique se os recursos de VM da tarefa de migração podem acessar essas instâncias.

  - Se as instâncias de Redis de origem e de destino estiverem na mesma região, crie uma conexão de emparelhamento VPC consultando **Conexão de emparelhamento de VPC**.
  - Se as instâncias de Redis de origem e de destino estiverem em regiões diferentes, crie uma conexão de nuvem consultando **Primeiros passos da Cloud Connect**.

- As instâncias de destino e de origem usam a mesma porta.
- A comutação de 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 obter detalhes, consulte **Modos de migração de dados de DCS**.
  - A tabela seguinte lista os cenários de comutação IP suportados.

Tabela 1-6 Cenários de comutação IP

Fonte	Alvo
Nó único, divisão de leitura/gravação ou principal/em espera	Nó único, divisão de leitura/gravação, principal/em espera ou Proxy Cluster
Proxy Cluster	Nó único, divisão de leitura/gravação, principal/em espera ou Proxy Cluster

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

1.7 Quais são as especificações de CPU das instâncias de DCS?

Ao usar a edição básica do DCS for Redis, você só precisa prestar atenção a indicadores críticos, como QPS, largura de banda e memória. Você não precisa se preocupar com as especificações da CPU.

A edição básica do DCS for Redis é baseada no Redis de código aberto. O Redis de código aberto usa um único thread principal para processar comandos, portanto, apenas um núcleo de CPU é usado em cada nó do Redis.

Devido a essa restrição, você pode usar um cluster e adicionar partições para obter maior desempenho da CPU. Cada nó em uma instância de cluster é alocado um núcleo de CPU por padrão.

A edição profissional de DCS for Redis é multithreaded. Para obter detalhes sobre as especificações da CPU, consulte **Tabela 1-7**.

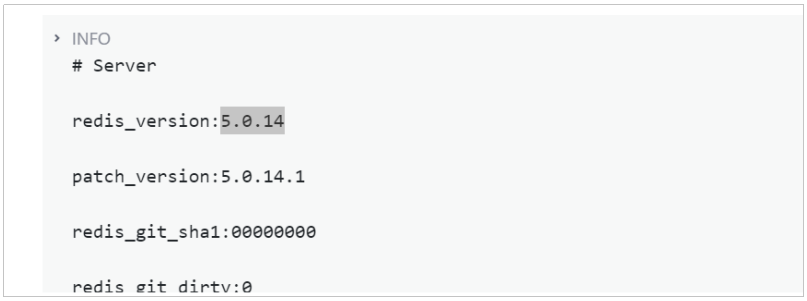
Tabela 1-7 Especificações de CPU do Redis de edição profissional

Memória de instância	CPU de instâncias principal/em espera de edição profissional (desempenho)	CPU de instâncias principal/em espera de edição profissional (armazenamento)
8 GB	4 vCPUs	4 vCPUs
16 GB	8 vCPUs	8 vCPUs
32 GB	8 vCPUs	16 vCPUs
64 GB	16 vCPUs	-

1.8 Como exibir a versão de uma instância do DCS Redis?

Conecte-se à instância e execute o comando **INFO**.

Figura 1-4 Consultar informações da instância



2 Cliente e conexão de rede

2.1 Como configurar um grupo de segurança?

As instâncias do DCS Redis 3.0/4.0/5.0/6.0 e do Memcached são implementadas 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 edição profissional, você pode usar grupos de segurança. Listas brancas não são suportadas. As operações de grupo de segurança são descritas nesta seção.
- Para controlar o acesso às instâncias do DCS Redis 4.0/5.0/6.0 edição básica, você pode usar listas brancas. Grupos de segurança não são suportados. As operações da lista branca são descritas em **Gerenciamento da lista branca de endereços IP**.

A seguir, descrevemos como configurar grupos de segurança para **intra-VPC access** e **public access** a instâncias do DCS Redis 3.0, Memcached e Redis 6.0 edição profissional.

Acesso dentro da VPC a instâncias do DCS Redis 3.0, Memcached e Redis 6.0 edição profissional

Um ECS pode se comunicar com uma instância de DCS se ela pertencer à mesma VPC e as mesmas regras de sub-rede e grupo de segurança estiverem configuradas corretamente.

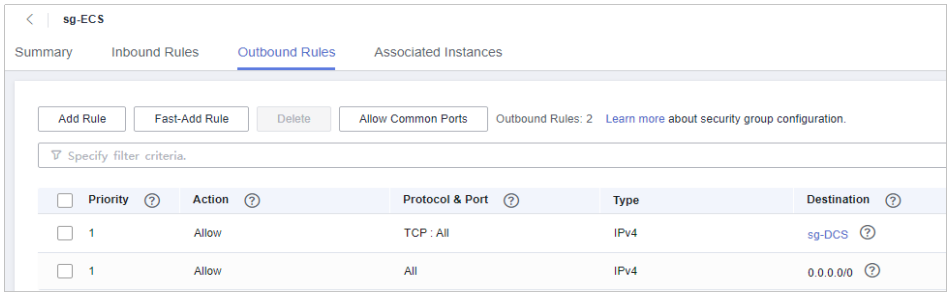
- Se a instância do ECS e do DCS estiverem configuradas com o mesmo grupo de segurança, o acesso à rede no grupo não será restrito por padrão.
- Se a instância do ECS e do DCS estiverem configuradas com grupos de segurança diferentes, adicione regras de grupo de segurança para garantir que a instância do ECS e do DCS possam acessar uma à outra.

  NOTA

  - Suponha que o ECS no qual o cliente é executado pertence ao grupo de segurança **sg-ECS** e a instância de DCS que o cliente acessará pertence ao grupo de segurança **sg-DCS**.
  - A seguir, a porta 6379 é usada para instâncias do DCS Redis 3.0 como exemplo. Para outras instâncias, use a porta real.
  - A extremidade remota é um grupo de segurança ou um endereço IP.

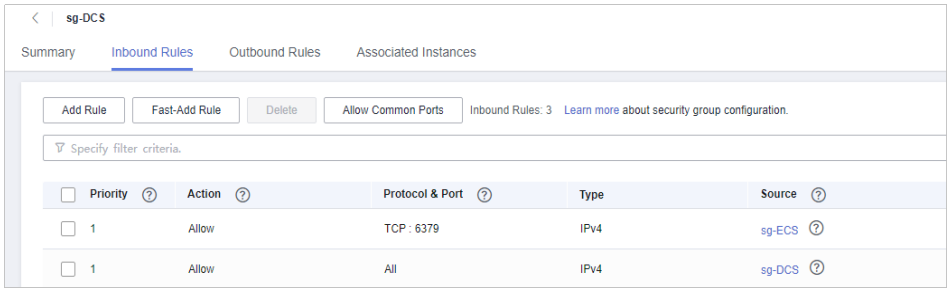
  a. Configurar o grupo de segurança para o ECS.

Adicione a seguinte regra de saída para permitir que o ECS acesse a instância do DCS. Ignore esta regra se não houver restrições no tráfego de saída.



b. Configurar o grupo de segurança para a instância do DCS.

Para garantir que seu cliente possa acessar a instância de DCS, adicione a seguinte regra de entrada ao grupo de segurança configurado para a instância de DCS:



AVISO

Para o endereço IP de origem, use o endereço IP especificado da instância do DCS. Evite usar **0.0.0.0/0** para evitar que ECSs vinculados ao mesmo grupo de segurança sejam atacados por explorações de vulnerabilidade do Redis.

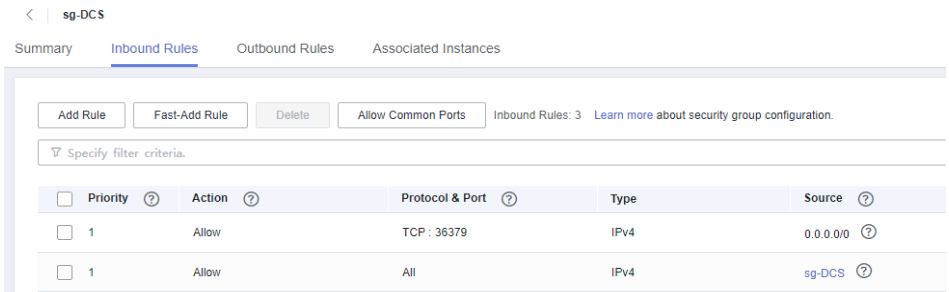
Acesso público a instâncias do DCS Redis 3.0

Um cliente pode acessar uma instância de DCS somente depois que as regras forem configuradas corretamente para o grupo de segurança da instância.

Por exemplo, para o grupo de segurança **sg-DCS**, você precisa configurar as seguintes regras na direção de entrada:

Defina o protocolo como TCP e o endereço IP de origem como 0.0.0.0/0 ou um endereço de cliente especificado. Se SSL estiver habilitado, defina o número da porta como 36379. Se o SSL estiver desabilitado, defina o número da porta como 6379. Veja a figura a seguir.

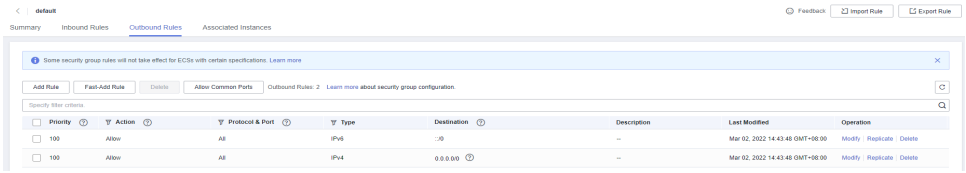
Figura 2-1 Regra de grupo de segurança (a porta 36379 é usada para o exemplo)



Grupo de segurança de uma tarefa de migração

- Ao criar uma tarefa de migração on-line, selecione um grupo de segurança. Suas regras de saída devem permitir o tráfego sobre os endereços IP e as portas das instâncias do Redis de origem e de destino. Por padrão, todo o tráfego de saída é permitido.
- A importação de backup usa o grupo de segurança **default**. Certifique-se de que todo o tráfego de saída seja permitido (esta é a configuração padrão)

Figura 2-2 Regras de saída do grupo de segurança de migração



2.2 O DCS suporta o acesso público?

- Redis 3.0

  Atualmente, **o acesso público é suportado apenas por instâncias do DCS Redis 3.0 protegidas por senha**. Você pode habilitar ou desabilitar o SSL para acesso público. É aconselhável fazer o download de um certificado de AC antecipadamente e usá-lo para verificar o certificado de uma instância de DCS para fins de segurança. Para obter mais informações, consulte **Acesso público a uma instância do DCS Redis**.

- Redis 4.0 e Redis 5.0

  O acesso público não é suportado pelas instâncias do DCS Redis 4.0, 5.0 e 6.0. 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 de DCS. Para obter detalhes, consulte **Uso do Nginx para acesso público a instâncias de DCS Redis de cluster de nó único, principal/em espera ou Proxy**.

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

  Você pode usar o Elastic Load Balance (ELB) para acessar diferentes tipos de instâncias de DCS em redes públicas. Para obter detalhes, consulte **Uso do ELB para acesso público ao DCS**.

- Memcached

  O acesso público não é suportado. O ECS que serve como cliente e a instância do DCS que o cliente acessará devem pertencer à mesma VPC. Nas fases de desenvolvimento e depuração da aplicação, você também pode usar o SSH para acessar sua instância no ambiente local. Para obter detalhes, consulte **Uso do túnel SSH para acesso público a uma instância do DCS**.

2.3 O DCS oferece suporte ao acesso entre VPCs?

Entre VPCs significa que o cliente e a instância não estão na mesma VPC.

O seguinte pressupõe que o acesso público está desabilitado para uma instância de DCS. Geralmente, as VPCs são isoladas umas das outras e um ECS não pode acessar uma instância do DCS que pertence a uma VPC diferente do ECS.

Você pode estabelecer conexões de emparelhamento de VPC para permitir que o ECS acesse a instância do DCS entre VPCs.

Ao usar conexões de emparelhamento de VPC para acessar instâncias de DCS entre VPCs, siga as seguintes regras:

- Se os blocos CIDR 172.16.0.0/12 a 172.16.0.0/24 forem usados durante a criação da instância de DCS, o cliente não poderá estar em nenhum dos seguintes blocos CIDR: 192.168.1.0/24, 192.168.2.0/24 e 192.168.3.0/24.
- Se os blocos CIDR 192.168.0.0/16 a 192.168.0.0/24 forem usados durante a criação da instância de DCS, o cliente não poderá estar em nenhum dos seguintes blocos CIDR: 172.31.1.0/24, 172.31.2.0/24 e 172.31.3.0/24.
- Se os blocos CIDR 10.0.0.0/8 a 10.0.0.0/24 forem usados durante a criação da instância de DCS, o cliente não poderá estar em nenhum dos seguintes blocos CIDR: 172.31.1.0/24, 172.31.2.0/24 e 172.31.3.0/24.

Para obter mais informações sobre como criar e usar conexões de emparelhamento de VPC, consulte **Conexão de emparelhamento de VPC**.

2.4 Serei cobrado pelo EIP usado para acesso público a uma instância do DCS Redis?

Sim. Você deve pagar pelo EIP usado para acesso público a uma instância do DCS Redis 3.0.

Antes de habilitar o acesso público, você deve ter um EIP disponível. Para obter os detalhes de cobrança, consulte os **Detalhes do preço do EIP**.

2.5 Por que "(error) NOAUTH Authentication required" é exibida quando eu acesso uma instância do DCS Redis?

Isso ocorre porque você habilitou o acesso sem senha para a instância. Para evitar que a mensagem de erro apareça, não insira nenhuma senha.

2.6 O que devo fazer se o acesso ao DCS falhar após a desconexão do servidor?

Análise: se as conexões persistentes ("pconnect" na terminologia do Redis) ou o pool de conexões forem usados e as conexões forem fechadas depois de serem usadas para conexão com instâncias do DCS, os erros serão retornados nas tentativas de reutilização das conexões.

Solução: ao usar o pconnect ou o pool de conexões, não encerre a conexão após o término de uma solicitação. Se a conexão for interrompida, restabeleça-a.

2.7 Por que as solicitações às vezes esgotam o tempo nos clientes?

Os erros ocasionais do tempo limite são normais devido à conectividade de rede e às configurações do tempo limite do cliente.

Recomendamos que você inclua operações de reconexão em seu código de serviço para evitar falhas de serviço se uma única solicitação falhar.

Se uma solicitação de conexão expirar, verifique se a persistência AOF foi ativada. Para evitar o bloqueio, certifique-se de que AOF foi ativado.

Se ocorrerem erros de tempo limite com frequência, entre em contato com o suporte técnico.

2.8 O que devo fazer se um erro for retornado quando eu usar o pool de conexão Jedis?

A mensagem de erro que possivelmente será exibida quando você usar o pool de conexão Jedis JedisPool é a seguinte:

```
redis.clients.jedis.exceptions.JedisConnectionException: Could not get a resource
from the pool
```

Se essa mensagem de erro for exibida, verifique se a instância está sendo executada corretamente. Se estiver funcionando corretamente, execute as seguintes verificações:

Passo 1 Verifique a rede.

1. Verifique as configurações de endereço IP.

   Verifique se o endereço IP configurado no cliente Jedis é o mesmo que o endereço de sub-rede configurado para sua instância de DCS. Se o acesso público estiver habilitado para sua instância, verifique se o endereço IP configurado no cliente Jedis é o mesmo que o EIP vinculado à sua instância. Se forem inconsistentes, modifique a configuração do endereço IP e tente novamente.

2. Teste a rede.

   Use o comando ping e telnet no cliente para testar a rede.

   - Se não for possível fazer ping na rede:
     - Para acesso dentro da VPC, verifique se o cliente e a sua instância do DCS estão na mesma VPC e se **as regras de grupo de segurança ou a lista branca** foram configuradas corretamente.
     - Para acesso público com SSL, verifique se você configurou o grupo de segurança de sua instância de DCS, permitindo o acesso por meio da porta 36379, conforme instruído em **Configurações de grupo de segurança**.
     - Para acesso público sem SSL, verifique se você configurou o grupo de segurança da instância do DCS, permitindo o acesso pela porta 6379, conforme instruído em **Configurações de grupo de segurança**.
   - Se for possível fazer o ping do endereço IP, mas o telnet falhar, reinicie sua instância. Se o problema persistir após a reinicialização, entre em contato com o suporte técnico.

Passo 2 Verifique o número de conexões.

Verifique se o número de conexões de rede estabelecidas excede o limite superior configurado para JedisPool. Se o número de conexões estabelecidas se aproximar do limite superior configurado, reinicie o serviço DCS e verifique se o problema persiste. Se o número de conexões estabelecidas estiver muito abaixo do limite superior, continue com as seguintes verificações.

No Unix ou Linux, execute o seguinte comando para consultar o número de conexões de rede estabelecidas:

netstat -an | grep 6379 | grep ESTABLISHED | wc -l

No Windows, execute o seguinte comando para consultar o número de conexões de rede estabelecidas:

netstat -an | find "6379" | find "ESTABLISHED" /C

Passo 3 Verifique o código JedisPool.

Se o número de conexões estabelecidas se aproximar do limite superior, determine se o problema é causado pela simultaneidade do serviço ou pelo uso incorreto de JedisPool.

Ao usar o JedisPool, você deve chamar **jedisPool.returnResource()** ou **jedis.close()** (recomendado) para liberar os recursos depois de chamar **jedisPool.getResource()**.

Passo 4 Verifique o número de conexões TIME_WAIT.

Execute o comando **ss -s** para verificar se há muitas conexões **TIME_WAIT** no cliente.

```
root@heru-nodelete:~# ss -s
Total: 140 (kernel 240)
TCP:   11 (estab 3, closed 1, orphaned 0, synrecv 0, timewait 0/0), ports 0

Transport Total     IP        IPv6
*         240       -         -
RAW       0         0         0
UDP       2         2         0
TCP       10        6         4
INET      12        8         4
FRAG      0         0         0
```

Se houver muitas conexões **TIME_WAIT**, modifique os parâmetros do kernel executando o comando **/etc/sysctl.conf** da seguinte maneira:

```
##Uses cookies to prevent some SYN flood attacks when the SYN waiting queue
overflows.
net.ipv4.tcp_syncookies = 1
##Reuses TIME_WAIT sockets for new TCP connections.
net.ipv4.tcp_tw_reuse = 1
##Enables quick reclamation of TIME_WAIT sockets in TCP connections.
net.ipv4.tcp_tw_recycle = 1
##Modifies the default timeout time of the system.
net.ipv4.tcp_fin_timeout = 30
```

Após a modificação, execute o comando **/sbin/sysctl -p** para que a modificação tenha efeito.

Passo 5 Se o problema persistir depois de executar as verificações anteriores, execute as etapas a seguir.

Capture pacotes e envie arquivos de pacotes juntamente com o tempo e a descrição da exceção ao suporte técnico para análise.

Execute o seguinte comando para capturar pacotes:

tcpdump -i eth0 tcp and port 6379 -n -nn -s 74 -w dump.pcap

No Windows, você também pode instalar a ferramenta Wireshark para capturar pacotes.

NOTA

Para acesso público, altere o número da porta para **36379**.

Substitua o nome da NIC pelo nome real.

----Fim

2.9 Como acessar uma instância do DCS Redis por meio do Redis Desktop Manager?

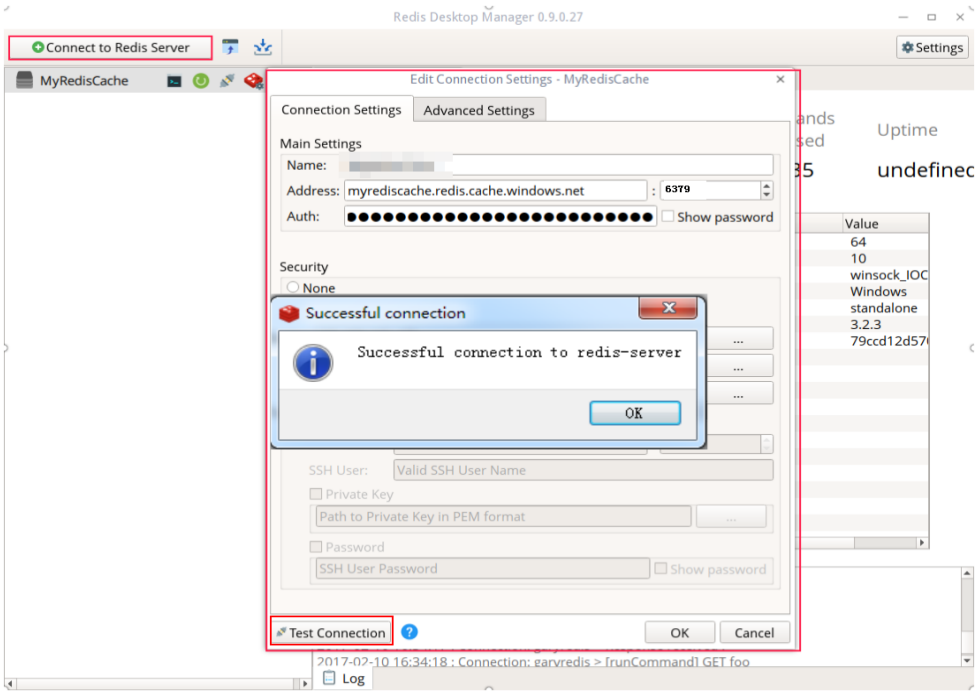
Você pode acessar uma instância do DCS Redis 3.0 por meio do Redis Desktop Manager em uma VPC ou pela Internet.

Dentro de uma VPC

1. Digite o endereço, o número da porta (6379) e a senha de autenticação da instância do DCS que você deseja acessar.
2. Clique em **Test Connection**.

   O sistema exibirá uma mensagem de sucesso se a conexão for bem-sucedida.

Figura 2-3 Acessar uma instância do DCS Redis por meio do Redis Desktop Manager na intranet



NOTA

Ao acessar uma instância de DCS de cluster, o comando Redis é executado corretamente, mas uma mensagem de erro pode ser exibida à esquerda porque os clusters de DCS são baseados no Codis, que difere do Redis nativo em termos da saída do comando **INFO**.

Pela Internet

Verifique se o SSL está habilitado para a instância do DCS que você deseja acessar.

- Se o SSL não estiver ativado, insira o endereço de acesso público da instância.

  Configure a regra de entrada do grupo de segurança da instância, permitindo o acesso pela porta 6379.

- Se o SSL estiver ativado, instale o cliente Stunnel e, em seguida, conecte-se ao servidor Redis por meio do Redis Desktop Manager. Observações:
  - O cliente Stunnel deve ser instalado. Para obter detalhes sobre como instalar e configurar o cliente Stunnel, consulte **instruções de Stunnel**.
  - O endereço deve ser definido como **127.0.0.1** em vez do endereço IP público. Caso contrário, a "connection reset" será retornada.

    Quando o SSL é ativado, o Redis é acessado por meio de um canal criptografado estabelecido pelo Stunnel. Depois que uma solicitação é enviada do Redis Desktop Manager para a porta de escuta de 127.0.0.1, a solicitação é criptografada e enviada para a instância do Redis por meio da porta 36379 em uma rede pública.

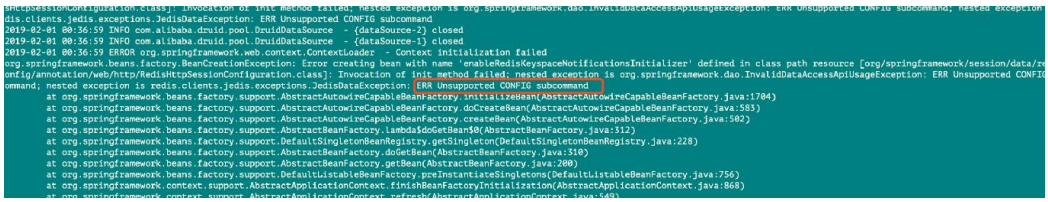
    Configure a regra de entrada do grupo de segurança da instância, permitindo o acesso pela porta 36379.

Para ativar o SSL, desative primeiro o acesso público. Em seguida, ative o SSL ao reativar o acesso público. Para desativar o SSL, desative primeiro o acesso público. Em seguida, desative o SSL ao reativar o acesso público.

2.10 O que acontece se "ERR Unsupported CONFIG subcommand" é exibido na SpringCloud?

Usando instâncias do DCS Redis, a Spring Session pode implementar o compartilhamento de sessão. Ao interconectar-se com a Spring Cloud, as seguintes informações de erro são exibidas:

Figura 2-4 Informações de erro da Spring Cloud



Por motivos de segurança, o DCS não oferece suporte ao comando **CONFIG** iniciado por um cliente. Você precisa executar as seguintes etapas:

1. No console do DCS, defina o valor do parâmetro **notify-keyspace-event** como **Egx** para uma instância do DCS Redis.
2. Adicione o seguinte conteúdo ao arquivo de configuração XML do framework Spring:

   <util:constant

   static-
   field="org.springframework.session.data.redis.config.ConfigureRedisAction.NO_OP"/>
3. Modifique o código Spring relacionado. Permita que o componente do bean **ConfigureRedisAction.NO_OP** proíba que um cliente invoque o comando **CONFIG**.

   @Bean

   public static ConfigureRedisAction configureRedisAction() {


```
return ConfigureRedisAction.NO_OP;
}
```

Para obter mais informações, consulte a **Documentação da Spring Session**.

AVISO

O compartilhamento de sessão é suportado apenas por instâncias do DCS Redis de **nó único** e **principal/em espera**, mas não por instâncias do DCS Redis de cluster.

2.11 O que posso fazer se não conseguir acessar uma instância de DCS usando seu endereço de nome de domínio?

Se um cliente não conseguir se conectar a uma instância de DCS usando o endereço de nome de domínio, defina o endereço do servidor DNS da sub-rede como o endereço do servidor DNS privado.

Para obter detalhes, consulte **Como mudar para um servidor DNS privado?**

2.12 É necessária uma senha para acessar uma instância? Como definir uma senha?

- Uma instância de DCS Redis pode ser acessada com ou sem senha. Você pode acessar diretamente uma instância do DCS Redis por meio de um cliente de Redis sem definir uma senha. No entanto, por motivos de segurança, é aconselhável definir uma senha para autenticação e verificação sempre que possível. A senha deve ser definida quando você criar a instância.
- Uma instância de DCS Memcached pode ser acessada com ou sem senha. Você pode selecionar qualquer cliente do Memcached que suporte o protocolo de texto e o protocolo binário do Memcached com base em recursos específicos da aplicação. A senha deve ser definida quando você criar a instância.
- Para alterar o modo de acesso à instância do Redis ou alterar ou redefinir uma senha, consulte **Gerenciamento de senhas**.

2.13 Posso acessar instâncias de DCS em um ambiente local?

- Se o acesso público estiver desativado para uma instância do DCS, você não poderá acessá-la em ambientes locais e só poderá acessá-la por meio de um ECS em uma VPC que possa se comunicar com a instância. As VPCs são usadas para garantir a segurança da rede dos serviços.

  Você pode se conectar a uma instância de DCS do seu ambiente local usando um ECS que pode se comunicar com sua instância para encaminhar suas solicitações. Para obter detalhes, consulte **Uso do túnel SSH para acesso público a uma instância do DCS**.

- Se o acesso público estiver habilitado, as instâncias do DCS poderão ser acessadas em ambientes locais. Para obter mais informações, consulte **Acesso público a uma instância de DCS Redis**.

2.14 O que deve ser observado ao usar o Redis para Pub/Sub?

O **site oficial do Redis** descreve o Pub/Sub em detalhes. Ao usar o Redis para Pub/Sub, observe o seguinte:

- Seu cliente deve processar as mensagens em tempo hábil.

  Seu cliente se inscreve em um canal. Se ele não receber mensagens em tempo hábil, as mensagens da instância do DCS podem estar sobrecarregadas. Se o tamanho das mensagens acumuladas atingir o limite (32 MB por padrão) ou permanecer em um determinado nível (8 MB por padrão) por um determinado período de tempo (1 minuto por padrão), seu cliente será desconectado automaticamente para evitar o esgotamento da memória do servidor.
- Seu cliente deve apoiar o restabelecimento da conexão em caso de desconexão.

  No caso de uma desconexão, você precisa executar o comando **subscribe** ou **psubscribe** no seu cliente para se inscrever em um canal novamente. Caso contrário, seu cliente não poderá receber mensagens.
- Não use pub/sub em cenários com altos requisitos de confiabilidade de mensagens.

  O Redis pub/sub não é um sistema de mensagens confiável. As mensagens que não forem recuperadas serão descartadas quando o cliente for desconectado ou ocorrer uma alternância principal/em espera.

2.15 Por que o acesso público à minha instância do DCS Redis foi desativado de forma não intencional?

Sintoma: o acesso público foi habilitado para uma instância do DCS Redis 3.0, mas é desabilitado repentinamente.

Motivo: o EIP vinculado à instância do DCS Redis não está vinculado. Como resultado, o acesso público é automaticamente desativado.

2.16 O que posso fazer se o erro "Cannot assign requested address" for retornado ao acessar o Redis usando o connect?

Sintoma

A mensagem de erro "Cannot assign requested address" é retornada quando você acessa o Redis usando **connect**.

Análise

Aplicações que encontram este erro normalmente usam php-fpm e phpredis. Em cenários de alta concorrência, um grande número de conexões TCP estão no estado TIME-WAIT. Como resultado, o cliente não pode alocar novas portas e a mensagem de erro será retornada.

Soluções

- Solução 1: use **pconnect** em vez de **connect**.

  O uso do **pconnect** reduz o número de conexões TCP e impede que as conexões sejam restabelecidas para cada solicitação e, portanto, reduz a latência.

  Ao usar o **connect**, o código para se conectar ao Redis é o seguinte:

  ```
  $redis->connect('${Hostname}',${Port});
  $redis->auth('${Inst_Password}');
  ```

  Substitua **connect** por **pconnect**, e o código se torna:

  ```
  $redis->pconnect('${Hostname}', ${Port}, 0, NULL, 0, 0, ['auth' => ['$
  {Inst_Password}']]);
  ```

  NOTA

  - Substitua os parâmetros de conexão no exemplo por valores reais. *${Hostname}*, *${Port}* e *${Inst_Password}* são o endereço de conexão, o número da porta e a senha da instância do Redis, respectivamente.
  - phpredis deve ser v5.3.0 ou posterior. Recomendamos que você use este modo de inicialização do **pconnect** para evitar erros NOAUTH durante a desconexão.

- Solução 2: modifique o parâmetro **tcp_max_tw_buckets** do ECS no qual o cliente está localizado.

  Nesta solução, as portas usadas pelas conexões TIME-WAIT são reutilizadas. No entanto, se a retransmissão ocorrer entre o ECS e o serviço de back-end, a conexão poderá falhar. Portanto, a solução **pconnect** é recomendada.

  a. Conecte-se ao ECS onde o cliente está localizado

  b. Execute o seguinte comando para verificar os parâmetros **ip_local_port_range** e **tcp_max_tw_buckets**:

  sysctl net.ipv4.tcp_max_tw_buckets net.ipv4.ip_local_port_range

  Informação semelhante à seguinte foi exibida:

  ```
  net.ipv4.tcp_max_tw_buckets = 262144
  net.ipv4.ip_local_port_range = 32768  61000
  ```

  c. Execute o seguinte comando para definir o parâmetro **tcp_max_tw_buckets** para um valor menor que o valor de **ip_local_port_range**:

  sysctl -w net.ipv4.tcp_max_tw_buckets=10000

Geralmente, a solução 1 é recomendada. Em cenários especiais (por exemplo, o código de serviço envolve muitos componentes e é difícil de mudar) a solução 2 pode ser usada para atender aos altos requisitos de simultaneidade.

2.17 Seleção de pool de conexão e configurações de parâmetro Jedis recomendadas

Vantagens de pool de conexão Jedis

A comparação entre Lettuce e Jedis é a seguinte:

- Lettuce
  - Lettuce não realiza a detecção de manutenção de atividade de conexão. Se existir uma conexão anormal no pool de conexões, um erro será relatado quando o tempo limite das solicitações.
  - Lettuce não implementa a validação do pool de conexões, como **testOnBorrow**. Como resultado, as conexões não podem ser validadas antes de serem usadas.
- Jedis
  - Jedis implementa a validação do pool de conexão usando **testOnBorrow**, **testWhileIdle** e **testOnReturn**.

    Se **testOnBorrow** estiver ativado, a validação de conexão será executada quando as conexões estiverem sendo emprestadas, o que tem a maior confiabilidade, mas afeta o desempenho (a detecção é realizada antes de cada solicitação do Redis).
  - **testWhileIdle** pode ser usado para detectar conexões ociosas. Se o limite for definido corretamente, as conexões anormais no pool de conexões poderão ser removidas a tempo de evitar erros de serviço causados por conexões anormais.
  - Se uma conexão se tornar anormal antes da verificação de conexão ociosa, o serviço que usa a conexão pode relatar um erro. Você pode especificar o parâmetro **timeBetweenEvictionRunsMillis** para controlar o intervalo de verificação.

Portanto, Jedis tem melhores capacidades de tratamento e detecção de exceções e é mais confiável do que Lettuce em cenários onde há exceções de conexão e jitters de rede.

Configurações de parâmetros recomendadas do pool de conexão Jedis

Tabela 2-1 Configurações recomendadas de parâmetros de pool de conexão Jedis

Parâmetro	Descrição	Configuração recomendada
maxTotal	Número máximo de conexões	Defina esse parâmetro com base no número de threads HTTP do contêiner da Web e das conexões reservadas. Supondo que o parâmetro **maxConnections** do Conector de Tomcat esteja definido como **150** e que cada solicitação HTTP possa enviar simultaneamente duas solicitações ao Redis, é recomendável definir esse parâmetro como pelo menos 400 (150 x 2 + 100). **Limite**: o valor de **maxTotal** multiplicado pelo número de nós do cliente (contêineres CCE ou VMs de serviço) deve ser menor que o número máximo de conexões permitidas para uma única instância do DCS Redis. Por exemplo, se **maxClients** de uma instância principal/em espera do DCS Redis for 10.000 e **maxTotal** de um único cliente for 500, o número máximo de clientes será 20.
maxIdle	Número máximo de conexões ociosas	Defina este parâmetro para o valor de **maxTotal**.

Parâmetro	Descrição	Configuração recomendada
minIdle	Número mínimo de conexões ociosas	Geralmente, é aconselhável definir este parâmetro para 1/X de **maxTotal**. Por exemplo, o valor recomendado é **100**. Em cenários sensíveis ao desempenho, você pode definir esse parâmetro para o valor de **maxIdle** para evitar o impacto causado por alterações frequentes na quantidade de conexão. Por exemplo, defina este parâmetro como **400**.
maxWaitMillis	Tempo máximo de espera para obter uma conexão, em milissegundos	O tempo de espera máximo recomendado para obter uma conexão do pool de conexões é o tempo limite máximo tolerável de um único serviço menos o tempo limite para a execução do comando. Por exemplo, se o tempo limite máximo tolerável de HTTP for 15s e o tempo limite de solicitações do Redis for 10s, defina esse parâmetro como 5s.
timeout	Tempo limite de execução do comando, em milissegundos	Esse parâmetro indica o tempo limite máximo para execução de um comando do Redis. Defina este parâmetro com base na lógica do serviço. Geralmente, é aconselhável definir esse tempo limite para mais de 210 ms para garantir a tolerância a falhas de rede. Para lógica de detecção especial ou detecção de exceção de ambiente, você pode ajustar esse tempo limite para segundos.

Parâmetro	Descrição	Configuração recomendada
minEvictableIdleTimeMillis	Tempo de despejo da conexão ociosa, em milissegundos. Se uma conexão não for usada por um período maior do que isso, ela será liberada.	Se você não quiser que o sistema restabeleça conexões desconectadas com frequência, defina esse parâmetro para um valor grande (xx minutos) ou defina esse parâmetro para **–1** e verifique as conexões ociosas periodicamente.
timeBetweenEvictionRunsMillis	Intervalo para detectar conexões ociosas, em milissegundos	O valor é estimado com base no número de conexões ociosas no sistema. Por exemplo, se esse intervalo for definido como 30s, o sistema detectará conexões a cada 30s. Se uma conexão anormal for detectada dentro de 30s, ela será removida. Defina este parâmetro com base no número de conexões. Se o número de conexões for muito grande e esse intervalo for muito curto, os recursos de solicitação serão desperdiçados. Se houver centenas de conexões, é aconselhável definir esse parâmetro para 30s. O valor pode ser ajustado dinamicamente com base nos requisitos do sistema.
testOnBorrow	Indica se a validade da conexão deve ser verificada usando o comando **ping** ao emprestar conexões do pool de recursos. Conexões inválidas serão removidas.	Se o serviço for extremamente sensível a conexões e o desempenho for aceitável, você poderá definir esse parâmetro como **True**. Geralmente, é aconselhável definir esse parâmetro como **False** para habilitar a detecção de conexão ociosa.

Parâmetro	Descrição	Configuração recomendada
testWhileIdle	Indica se o comando **ping** deve ser usado para monitorar a validade da conexão durante o monitoramento de recursos ociosos. Conexões inválidas serão destruídas.	Verdadeiro
testOnReturn	Indica se deve verificar a validade da conexão usando o comando **ping** ao retornar conexões ao pool de recursos. Conexões inválidas serão removidas.	Falso
maxAttempts	Número de tentativas de conexão quando o JedisCluster é usado	Valor recomendado: 3–5. Valor padrão: **5**. Defina esse parâmetro com base nos intervalos máximos de tempo limite das APIs de serviço e em uma única solicitação. O valor máximo é **10**. Se o valor exceder **10**, o tempo de processamento de uma única solicitação é muito longo, bloqueando outras solicitações.

2.18 O que fazer se um cliente Lettuce 6.x for incompatível com minha instância de DCS?

Sintoma

Quando um cliente Lettuce 6.x é conectado a uma instância do DCS Redis 4.x/5.x do Proxy Cluster, a mensagem de erro "NOAUTH Authentication required" é exibida.

Figura 2-5 Exemplo de mensagem de erro

```
[2022-01-04 18:33:35.219] [lettuce-nioEventLoop-4-1] [DEBUG] [io.lettuce.core.AbstractRedisClient:?] - Connecting to Redis at 192.168.xxx.xxx:6379, initialization
java.util.concurrent.CompletionException: io.lettuce.core.RedisCommandExecutionException: NOAUTH Authentication required.
    at java.util.concurrent.CompletableFuture.encodeThrowable(CompletableFuture.java:292)
    at java.util.concurrent.CompletableFuture.completeThrowable(CompletableFuture.java:308)
```

Análise

No Lettuce 6.x e versões posteriores, o comando **HELLO** do RESP3 (introduzido no Redis 6.x) é usado para determinar a adaptação da versão. Instâncias de versões anteriores que não suportam o comando **HELLO** podem encontrar problemas de compatibilidade. Para essas instâncias, você pode especificar o modo RESP2 (compatível com as versões 4 e 5 do Redis) em Lettuce.

Solução

Adicione o seguinte código para usar o protocolo RESP2 para acessar o Redis:

```
package com.chinaroad.parking.config;

import io.lettuce.core.ClientOptions;
import io.lettuce.core.protocol.ProtocolVersion;
import
org.springframework.boot.autoconfigure.data.redis.LettuceClientConfigurationBuilde
rCustomizer;
import org.springframework.context.annotation.Configuration;
import
org.springframework.data.redis.connection.lettuce.LettuceClientConfiguration;

@Configuration
public class SpringConfig implements LettuceClientConfigurationBuilderCustomizer {

    @Override
    public void
customize(LettuceClientConfiguration.LettuceClientConfigurationBuilder
clientConfigurationBuilder) {
        // manually specifying RESP2
        clientConfigurationBuilder.clientOptions(ClientOptions.builder()
                .protocolVersion(ProtocolVersion.RESP2)
                .build());
    }

}
```

2.19 Devo usar um nome de domínio ou um endereço IP para conectar-se a uma instância do DCS Redis?

- Cluster de nó único e Proxy:

  Cada instância tem apenas um endereço IP e um endereço de nome de domínio. Os endereços permanecem inalterados antes e depois da alternância principal/em espera. Você pode usar qualquer endereço para se conectar à instância.

- Principal/em espera (edição básica):

  Cada instância tem um endereço IP e dois endereços de nome de domínio. Um dos endereços de nome de domínio é usado apenas para processar solicitações de leitura. Os endereços permanecem inalterados após a alternância principal/em espera. Você pode usar qualquer endereço para se conectar à instância.

  Quando você usa um endereço de nome de domínio, faça a distinção entre solicitações de leitura e gravação. Se você usar **Connection Address** ou **IP Address**, as funções não serão afetadas. Se você usar **Read-only Address**, somente as solicitações de leitura serão processadas. Recomendamos que você use instâncias de divisão de leitura/gravação se tiver requisitos de divisão de leitura/gravação.

- Redis 6.0 (edição profissional)

  Use o nome de domínio para conexão. Pode haver vários endereços IP ou eles podem mudar.

- Redis Cluster:

  Uma instância do Redis Cluster tem vários pares de endereços IP principais e de réplica e um endereço de nome de domínio. Você pode usar qualquer endereço para se conectar à instância.

O nó conectado envia solicitações para o nó correto. Todos os nós no cluster podem receber solicitações. **Configure vários ou todos os endereços IP** para evitar pontos únicos de falha.

NOTA

- Os nomes de domínio não podem ser resolvidos entre regiões. Se o cliente e a instância do DCS Redis não estiverem na mesma região, a instância não poderá ser acessada usando 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 usando seu endereço IP. Para detalhes, veja **Restrições**.
- Para obter detalhes sobre como se conectar a uma instância, consulte **Acesso a uma instância do DCS Redis**.

2.20 O endereço somente leitura de uma instância principal/em espera está conectado ao nó principal ou em espera?

Uma instância do DCS Redis 4.0/5.06.0 básica principal/em espera tem um **Connection Address** e um **Read-only Address**. O endereço de conexão é usado para se conectar ao nó principal da instância e o endereço somente leitura é usado para se conectar ao nó em espera da instância.

Para obter detalhes, consulte **Arquitetura das instâncias de DCS Redis 4.0/5.0/6.0 edição básica principal/em espera**.

Figura 2-6 Endereços de instância

3 Uso do Redis

3.1 O que é memória reservada? Como configurar a memória reservada?

Memória reservada

A memória reservada é parte da memória que não é usada para armazenar dados, mas para persistência de dados, sincronização principal/em espera e backup.

Parâmetro **reserved-memory-percent** é usado para configurar a memória reservada.

AVISO

Nos dados de monitoramento, o uso de memória não inclui o uso de memória reservada.

Somente as instâncias a seguir devem ter memória reservada:

- Instâncias do DCS Redis 3.0 de nó único
- Instâncias principal/em espera do DCS Redis 3.0
- Instâncias de DCS Memcached de nó único
- Instâncias de DCS Memcached principal/em espera

Se a memória reservada for insuficiente porque os dados ocupam demasiada memória, poderão ocorrer os seguintes problemas:

- As operações na instância do DCS tornam-se lentas. (O sistema permite a troca, deteriorando o desempenho.)
- Não é possível fazer backup dos dados.
- Os dados não podem ser sincronizados entre os nós principais e em espera no tempo.
- As especificações da instância não podem ser alteradas.
- O processo pode ser reiniciado.

Procedimento para configurar a memória reservada

Altere o valor de **reserved-memory-percent** consultando **Modificação dos parâmetros de configuração de uma instância**.

NOTA

- Defina o parâmetro para pelo menos **30**. Para instâncias criadas em ou após 2021, o valor padrão é **30**.
- A porcentagem leva o máximo de memória disponível, em vez da memória total, como o todo. A memória disponível é listada na coluna **Available Memory** em **Especificações da instância de DCS**.

3.2 O que são quantidades de partições e réplicas?

Partição

Uma **shard** é uma unidade de gerenciamento em clusters do Redis. Cada partição corresponde a um processo redis-server. Um cluster consiste em vários partições. Cada partição tem vários slots. Os dados são armazenados de forma distribuída nos slots. Partições aumentam a capacidade de cache e conexões simultâneas.

Cada instância de cluster consiste em várias partições. Por padrão, cada partição é uma instância principal/em espera com duas réplicas. O número de partições é igual ao número de nós principais em uma instância de cluster.

Réplicas

Uma réplica refere-se a um **node** de uma instância de DCS. Pode ser um nó principal ou um nó em espera. Uma instância de réplica única não tem nenhum nó em espera. Uma instância de duas réplicas tem um nó principal e um nó em espera. Por exemplo, se o número de réplicas for definido como três para uma instância de nó principal/em espera, a instância terá um nó principal e dois nós em espera.

Número de réplicas e partições de diferentes tipos de instância

- **Nó único**: cada instância tem apenas um nó (um processo do Redis). Se o processo do Redis estiver com defeito, o DCS iniciará um novo processo do Redis para a instância.
- **Divisão principal/em espera e de leitura/gravação**: cada instância tem uma partição, que contém um nó principal e um ou mais nós em espera. Se o nó principal estiver defeituoso, a alternância principal/em espera será acionada para restaurar os serviços. Quanto mais as réplicas (nós em espera), melhor a confiabilidade (o desempenho não é afetado).
- **Cluster**: cada instância tem várias partições. Por padrão, cada partição é uma instância principal/em espera com duas réplicas. Por exemplo, se uma instância de cluster tiver três partições e três réplicas, cada partição terá três nós (um nó principal e dois nós em espera).

Tipos de instância	Partições	Réplicas	Balanceamento de carga	Endereços de IP
Nó único	1	-	-	1

Tipos de instância	Partições	Réplicas	Balanceamento de carga	Endereços de IP
Principal/em espera (edição básica)	1	Padrão: 2; personalizável: múltipla	Não suportado	O mesmo que o número de réplicas
Principal/em espera (edição profissional)	1	2 (não personalizável)	Não suportado	O mesmo que o número de réplicas
Divisão de leitura/ gravação	1	Padrão: 2; personalizável: múltipla	Suportado	1
Proxy Cluster	Múltiplas	2 (não personalizável)	Suportado	1
Redis Cluster	Múltiplas	Padrão: 2; personalizável: uma ou múltiplas	Não suportado	Número de réplicas x Número de partições

3.3 Posso alterar a VPC e a sub-rede de uma instância do DCS Redis?

Não. Depois que uma instância é criada, sua VPC e sub-rede não podem ser alteradas. Se você quiser usar um conjunto diferente de VPC e sub-rede, crie uma mesma instância e especifique um conjunto desejado de VPC e sub-rede. Depois que a nova instância for criada, você poderá migrar dados da instância antiga para a nova instância seguindo as **instruções de migração de dados**.

3.4 Por que os grupos de segurança não podem ser configurados para instâncias do DCS Redis 4.0/5.0/6.0 edição básica?

Atualmente, as instâncias do DCS Redis 4.0/5.0/6.0 edição básica usam pontos de extremidade de VPC e não oferecem suporte a grupos de segurança.

Para permitir o acesso somente de endereços IP específicos a uma instância do DCS Redis 4.0, 5.0 ou 6.0 edição básica, adicione os endereços IP à lista branca da instância.

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 3-1 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

3.5 As instâncias do DCS Redis limitam o tamanho de uma chave ou valor?

- O tamanho máximo permitido de uma chave é 512 MB.

  Para reduzir o uso de memória e facilitar a consulta de chaves, certifique-se de que cada chave não exceda 1 KB.
- O tamanho máximo permitido de uma cadeia é 512 MB.
- O tamanho máximo permitido de um Conjunto, Lista ou Hash é 512 MB.

  Em essência, um Conjunto é uma coleção de Strings; uma Lista é uma lista de Strings; um Hash contém mapeamentos entre campos de cadeia e valores de cadeia.

Evite que o cliente grave constantemente valores grandes no Redis. Caso contrário, a eficiência da transmissão da rede será reduzida e o servidor Redis levará mais tempo para processar comandos, resultando em maior latência.

3.6 Posso obter os endereços dos nós em uma instância do DCS Redis de cluster?

As instâncias do Redis 3.0 do DCS de cluster (tipo de Proxy Cluster) são usadas da mesma maneira que você usa instâncias de nó único ou principais/em espera. Você não precisa saber os endereços do nó de back-end.

Para uma instância de cluster DCS Redis 4.0 ou posterior (tipo de cluster Redis), execute o comando **CLUSTER NODES** para obter endereços de nó:

redis-cli -h *{redis_address}* -p *{redis_port}* -a *{redis_password}* cluster nodes

Na saída semelhante à seguinte, obtenha os endereços IP e os números de porta de todos os nós principais.

```
[root@ecs-[redacted]54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a [redacted]23 cluster nodes
fb75f0743af4695a3d241ff7790b2f508e4985ff 192.168.0.140:6379@16379 myself,master - 0 1562144170000 3 connected
d112bae791b2bbd9602fe32963536b8a0db9eb79 192.168.0.61:6379@16379 master - 0 1562144171524 1 connected 0-5460
73e2f8fe196166f9ad1283361867d24c136413f0 192.168.0.194:6379@16379 master - 0 1562144170000 2 connected 5461-10
40d72299fde6045de0f79ee4b97910b505acbc6a 192.168.0.231:6379@16379 slave 73e2f8fe196166f9ad1283361867d24c13641
be6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985f
c16b9acaeed7dd0721f129596cd43bd499c0e396 192.168.0.169:6379@16379 slave d112bae791b2bbd9602fe32963536b8a0db9eb
```

3.7 Por que a memória disponível é menor que o tamanho do cache de instância?

As instâncias do DCS Redis 3.0 e do Memcached são implementadas em VMs, portanto, uma pequena quantidade de memória é reservada para sobrecargas do sistema. Esse problema não ocorrerá em outras versões de instância.

3.8 O DCS for Redis suporta divisão de leitura/gravação?

A tabela a seguir descreve o suporte do DCS para divisão de leitura/gravação.

Tipos de instância	Divisão de leitura/gravação
Divisão de leitura/gravação	Compatível. **NOTA** Para implementar a divisão de leitura/gravação sem configurações de cliente, use **read/write splitting instances**.
Redis Cluster	A divisão de leitura/gravação pode ser configurada e implementada no cliente. Para mais detalhes, consulte **Configuração**.
Principal/em espera (Redis 4.0/5.0/6.0 básico)	A divisão de leitura/gravação pode ser implementada em um cliente que é capaz de distinguir entre solicitações de leitura e gravação.
Outros	Incompatível.

Configuração

- Para uma **Redis Cluster instance**, você pode consultar todos os nós principais e de réplica executando o comando **CLUSTER NODES**. O cliente se conectará às réplicas e configurará o acesso somente leitura nelas.

  Execute o seguinte comando para consultar nós de cluster:

  ```
  redis-cli -h {redis_address} -p {redis_port} -a {redis_password} cluster nodes
  ```

  A configuração somente leitura nas réplicas é obtida por meio do comando **READONLY**.
- Para uma **instância básica principal/em espera do DCS Redis 4.0/5.0/6.0**, há dois nomes de domínio exibidos na página de detalhes da instância do console: um endereço de leitura/gravação (nó principal) e um endereço somente leitura (nó em espera). No cliente, você pode direcionar solicitações de gravação para o nome de domínio de leitura/gravação e solicitações de leitura para o nome de domínio somente leitura.
- Para uma **read/write splitting instance**, a divisão de leitura/gravação é implementada no lado do servidor por padrão. Os proxies distinguem entre solicitações de leitura e gravação e encaminham solicitações de gravação para o nó principal e solicitações de leitura para o nó em espera. Você não precisa executar nenhuma configuração no cliente.

3.9 O DCS for Redis oferece suporte a vários bancos de dados?

O suporte do DCS para vários bancos de dados (multi-BD) é o seguinte:

- Instâncias de DCS Redis de nó único e principais/em espera: multi-BD é suportado. Por padrão, existem 256 bancos de dados, com numeração de 0 a 255. O banco de dados padrão é DB0. Multi-BD é usado para isolamento de dados. O tamanho de cada banco de dados não é alocado uniformemente. Como resultado, um banco de dados pode ocupar totalmente a memória da instância.
- Proxy Cluster: há apenas um banco de dados por padrão.
  - Para obter detalhes sobre como comprar uma instância de Proxy Cluster com vários bancos de dados, consulte **Como comprar uma instância de Proxy Cluster de vários bancos de dados?**

  - Para obter detalhes sobre como habilitar vários bancos de dados para uma instância de Proxy Cluster de banco de dados único, consulte **Quais são as restrições na implementação de vários bancos de dados em uma instância de Proxy Cluster?**

    NOTA

    As instâncias do DCS Redis 3.0 do Proxy Cluster não oferecem suporte a vários bancos de dados.

- Instâncias de DCS do Redis Cluster: multi-BD não é suportado. Existe apenas um banco de dados.

O número de bancos de dados não pode ser alterado e o tamanho de cada banco de dados não pode ser personalizado.

3.10 Como sei se uma instância é de banco de dados único ou de vários bancos de dados?

Divisão de nó único, principal/em espera e leitura/gravação: multi-BD (256 BDs, numerados de 0 a 255)

Proxy Cluster: banco de dados único por padrão. Multi-BD pode ser ativado. Para obter detalhes, consulte **Quais são as restrições na implementação de vários bancos de dados em uma instância de Proxy Cluster?**

Redis Cluster: banco de dados único. Multi-BD não é suportado.

Você pode se conectar a uma instância do DCS Redis 4.0 ou posterior no console para verificar se ela é multi-BD.

Figura 3-1 Conectar-se ao Redis

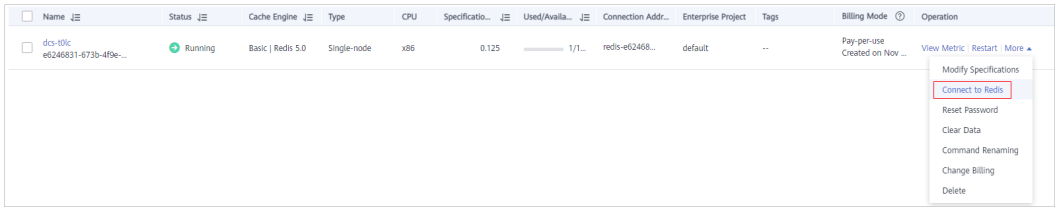
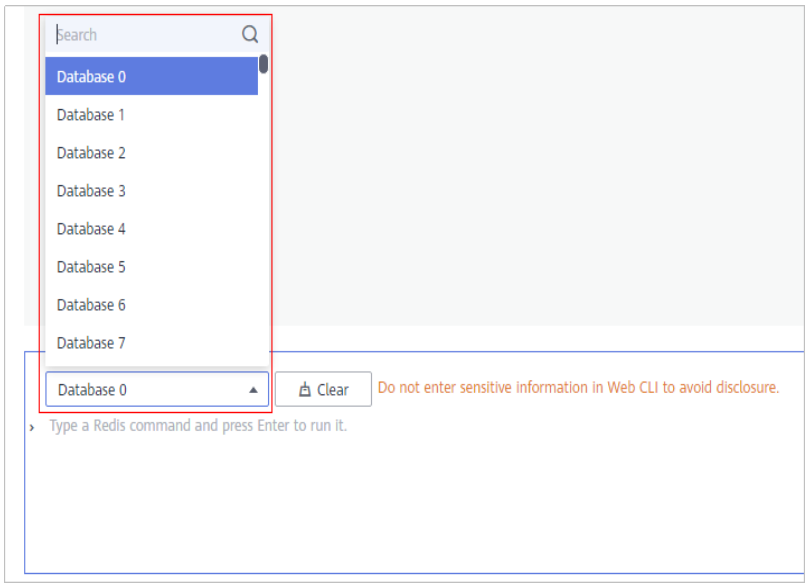


Figura 3-2 Exibição de bancos de dados



3.11 O DCS for Redis oferece suporte a clusters do Redis?

Sim. DCS for Redis 4.0 e 5.0 oferecem suporte a clusters de Proxy e clusters de Redis. O DCS for Redis 3.0 oferece suporte a clusters de Proxy. O DCS for Redis 6.0 oferece suporte a clusters do Redis.

3.12 O DCS for Redis oferece suporte a Sentinels?

- As instâncias de cluster e as instâncias principais/em espera do DCS Redis 4.0, 5.0 e 6.0 são compatíveis com Sentinels. Os Sentinels monitoram o status de execução dos nós principais e em espera de uma instância principal/em espera e de cada partição de uma instância de cluster. Se o nó principal se tornar defeituoso, um failover será executado.

  As Sentinels são invisíveis para você e são usadas apenas no serviço.
- O DCS for Redis 3.0 não suporta o Redis Sentinel. Em vez disso, ele usa keepalive para monitorar nós principais e de réplica e gerenciar failovers.

3.13 Qual é a política padrão de despejo de dados?

Os dados são removidos do cache com base em um limite de espaço definido pelo usuário para liberar espaço para novos dados. Para mais detalhes, visite o **site oficial do Redis**. Nas versões atuais do DCS for Redis, você pode selecionar uma política de despejo que preferir.

Você pode alterar a política de despejo configurando o parâmetro **maxmemory-policy**.

Quando **maxmemory** for atingido, você poderá selecionar uma das oito políticas de despejo a seguir:

- **noeviction**: quando o limite de memória é atingido, as instâncias de DCS retornam erros aos clientes e não processam mais solicitações de gravação e outras solicitações que podem resultar em mais memória a ser usada. No entanto, **DEL** e mais algumas solicitações de exceção podem continuar a ser processadas.

- **allkeys-lru**: as instâncias de DCS tentam remover primeiro as chaves usadas menos recentemente, a fim de liberar espaço para novos dados.
- **volatile-lru**: as instâncias de DCS tentam remover as chaves usadas menos recentemente com um conjunto de expiração primeiro, a fim de liberar espaço para novos dados.
- **allkeys-random**: as instâncias do DCS reciclam chaves aleatórias para que novos dados possam ser armazenados.
- **volatile-random**: as instâncias de DCS despejam chaves aleatórias com um conjunto de expiração, a fim de liberar espaço para novos dados.
- **volatile-ttl**: as instâncias de DCS removem as chaves com um conjunto de expiração e tentam remover as chaves com um tempo de vida (TTL) menor primeiro, a fim de liberar espaço para novos dados.
- **allkeys-lfu**: as instâncias do DCS removem as chaves usadas com menos frequência de todas as chaves.
- **volatile-lfu**: as instâncias do DCS removem as chaves usadas com menos frequência com um campo **expire** de todas as chaves.

NOTA

- Se nenhuma chave puder ser reciclada, **volatile-lru**, **volatile-random** e **volatile-ttl** são os mesmos que **noeviction**. Para obter detalhes, consulte a descrição de **noeviction**.
- A política de despejo padrão é **volatile-lru** para instâncias do DCS Redis criadas em ou após julho de 2020. A política de despejo padrão é **noeviction** para instâncias do DCS Redis criadas antes de julho de 2020.

3.14 O que devo fazer se ocorrer um erro no redis_exporter?

Inicie redis_exporter usando a CLI. Com base na saída, verifique se há erros e solucione problemas de acordo.

```
[root@ecs-swk /]./redis_exporter -redis.addr 192.168.0.23:6379
INFO[0000] Redis Metrics Exporter V0.15.0   build date:2018-01-19-04:08:01 sha1:
a0d9ec4704b4d35cd08544d395038f417716a03a
  Go:go1.9.2
INFO[0000] Providing metrics at :9121/metrics
INFO[0000] Connecting to redis hosts: []string{192.168.0.23:6379}
INFO[0000] Using alias:[]string{""}
```

3.15 Como proteger minhas instâncias do DCS Redis?

O Redis é uma das tecnologias de cache de código aberto mais poderosas e amplamente utilizadas. No entanto, o Redis de código aberto não possui recursos de segurança robustos próprios. É vulnerável a ataques maliciosos da Internet, possivelmente causando violações de dados.

Para proteger suas instâncias do DCS Redis, considere seguir os seguintes conselhos:

- Configurações de conexão de rede
  a. Criptografe dados confidenciais e desative o acesso público.

     Os dados confidenciais devem ser criptografados antes de serem armazenados. Não use o acesso público, a menos que seja exigido de outra forma.

b. Configure regras de acesso para os grupos de segurança.

Grupos de segurança e VPCs são projetados para proteger o acesso à rede. Permitir o acesso pelo menor número possível de portas para evitar riscos.

c. Configure firewalls do ECS.

Configure regras de filtragem de firewall para o ECS em que o cliente é executado.

d. Defina a senha da instância.

e. Configure uma lista branca.

- Utilização do redis-cli

a. Esconda a senha.

Problema: se a opção **-a <password>** for usada, a senha poderá aparecer quando o comando **ps** for executado.

Solução: modifique o código fonte do Redis. Esconda a senha imediatamente após iniciar o redis-cli chamando a função principal.

b. Desative o sudo em execução de scripts.

Problema: os parâmetros para iniciar o redis-cli contêm padrões sensíveis relacionados à senha, que podem aparecer quando o comando **ps** é executado e podem ser registrados.

Solução: acesse a instância chamando APIs (ou através do redis-py em Python). Não permitir alternar para o usuário **dbuser** usando sudo no redis-cli.

3.16 Por que o bloqueio distribuído do redisson não é suportado pelas instâncias do DCS Redis 3.0 de Proxy Cluster?

Redisson implementa a aquisição e o desbloqueio de bloqueios no seguinte processo:

1. A aquisição e o desbloqueio de bloqueios do redisson são implementados pela execução de scripts Lua.
2. Durante a aquisição de bloqueio, os comandos **EXISTS**, **HSET**, **PEXPIRE**, **HEXISTS**, **HINCRBY**, **PEXPIRE** e **PTTL** devem ser executados no script Lua.
3. Durante o desbloqueio, os comandos **EXISTS**, **PUBLISH**, **HEXISTS**, **PEXIPRE** e **DEL** devem ser executados no script Lua.

Em um cluster baseado em proxy, o proxy processa os comandos **PUBLISH** e **SUBSCRIBE** e encaminha solicitações para o servidor Redis. O comando **PUBLISH** não pode ser executado no script Lua.

Como resultado, as instâncias do DCS Redis 3.0 de Proxy Cluster não suportam bloqueios distribuídos do redisson. **Para usar o redisson, recorra ao Redis 4.0 ou 5.0**.

3.17 Posso personalizar ou alterar a porta para acessar uma instância de DCS?

Não é possível personalizar ou alterar a porta para acessar uma instância do DCS Redis 3.0 ou do Memcached. Você pode personalizar e alterar a porta para acessar uma instância do DCS Redis 4.0, 5.0 ou 6.0.

- Redis 3.0

  Acesso dentro de VPC: porta 6379; acesso público sem SSL: porta 6379; acesso público com SSL: porta 36379.

- Memcached

  Use a porta 11211 para acesso dentro de VPC. O acesso público não é suportado.

- Redis 4.0/5.0/6.0

  Você pode especificar uma porta (variando de 1 a 65535) ou usar a porta padrão (6379) para acessar uma instância do DCS Redis 4.0, 5.0 ou 6.0. Se nenhuma porta for especificada, a porta padrão será usada.

  O acesso público não é suportado pelas instâncias do DCS Redis 4.0/5.0/6.0.

Se a instância e o cliente estiverem em grupos de segurança diferentes, você deverá configurar regras de acesso para os grupos de segurança, permitindo o acesso por meio da porta especificada. Para mais detalhes, consulte **Como configurar um grupo de segurança?**.

Personalizar uma porta

Ao criar uma instância do DCS Redis 4.0, 5.0 ou 6.0, você pode inserir um número de porta para **IP Address**. Se você não especificar uma porta, a porta padrão 6379 será usada.

Mudar a porta

Depois que uma instância do DCS Redis 4.0, 5.0 ou 6.0 é criada, você pode alterar sua porta.

1. No painel de navegação do console de DCS, escolha **Cache Manager**.
2. Clique em uma instância do DCS Redis.
3. Na área **Connection**, clique em  ao lado de **Connection Address**.

AVISO

Depois que a porta é alterada, todas as conexões com a instância do Redis são interrompidas e os serviços são conectados à nova porta.

3.18 Posso modificar os endereços de conexão para acessar uma instância de DCS?

Depois que uma instância de DCS é criada, seu endereço IP e nome de domínio para acesso dentro da VPC não podem ser modificados. Se o acesso público tiver sido ativado para a instância, o endereço IP elástico (EIP) vinculado à instância poderá ser modificado.

Para usar um endereço IP diferente, você deve criar uma nova instância e especificar manualmente um endereço IP. Depois que a instância for criada, migre os dados da instância antiga para a nova instância.

3.19 Por que não consigo excluir uma instância?

Possíveis causas e soluções:

- A instância não está no estado **Running**.

  Somente as instâncias no estado **Running** podem ser excluídas.
- Verifique se a instância não foi criada.

  Para excluir instâncias que não foram criadas, clique no número ao lado de **Instance Creation Failures** no console do DCS.

3.20 O DCS oferece suporte à implementação entre AZs?

As instâncias de DCS Redis principal/em espera, divisão de leitura/gravação e de cluster e as instâncias de DCS Memcached principal/em espera podem ser implementadas em zonas de disponibilidade (AZs).

- Se os nós de instância em uma AZ estiverem com defeito, os nós em outras AZs não serão afetados. O nó em espera torna-se automaticamente o nó principal para continuar a operar, garantindo a recuperação de desastres (DR).
- A implementação entre AZs não compromete a velocidade da sincronização de dados entre os nós principais e em espera.

3.21 Por que leva muito tempo para iniciar uma instância de DCS de cluster?

Possível causa: 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**.

Solução: comece a gravar dados em uma instância somente depois que a instância for iniciada.

3.22 O DCS for Redis fornece software de gerenciamento de back-end?

Não. Para consultar configurações do Redis e informações de uso, use redis-cli. Se você quiser monitorar métricas de instâncias do DCS Redis, acesse o console do Cloud Eye ou execute as seguintes operações.

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

3.23 Posso recuperar dados excluídos de uma instância de DCS?

Se você tiver feito backup da instância do DCS, poderá restaurar seus dados a partir do backup. No entanto, a restauração substituirá os dados gravados antes da restauração.

Você pode restaurar dados de backup em uma instância principal/em espera, cluster ou de divisão de leitura/gravação por meio de **Backups & Restorations** no console do DCS. Para obter detalhes, consulte **Restauração de uma instância de DCS**.

Se uma instância de DCS for excluída, os dados da instância e seu backup também serão excluídos. Antes de excluir uma instância, você pode fazer o download dos arquivos de backup da instância para armazenamento local permanente e também pode migrá-los para uma nova instância se precisar restaurar os dados. Para obter detalhes sobre como baixar os dados de backup, consulte **Como exportar dados de instância do DCS Redis?**

3.24 A DCS for Redis oferece suporte à transmissão criptografada SSL?

Por padrão, o SSL está desabilitado para instâncias da edição básica de DCS Redis 6.0. Para ativá-lo, consulte .

Para acesso público a instâncias do DCS (suportado apenas por instâncias do DCS Redis 3.0), você pode ativar a criptografia TLS com Stunnel. Para obter detalhes, consulte as **instruções sobre como instalar e configurar o Stunnel**. Quando o DCS provisiona instâncias, a CA (Cadeia de certificados) especificada gera um certificado de serviço exclusivo para cada instância. Ao se conectar a uma instância, os clientes podem usar os certificados raiz da CA baixados do console de gerenciamento para autenticar o servidor da instância e criptografar os dados durante a transmissão.

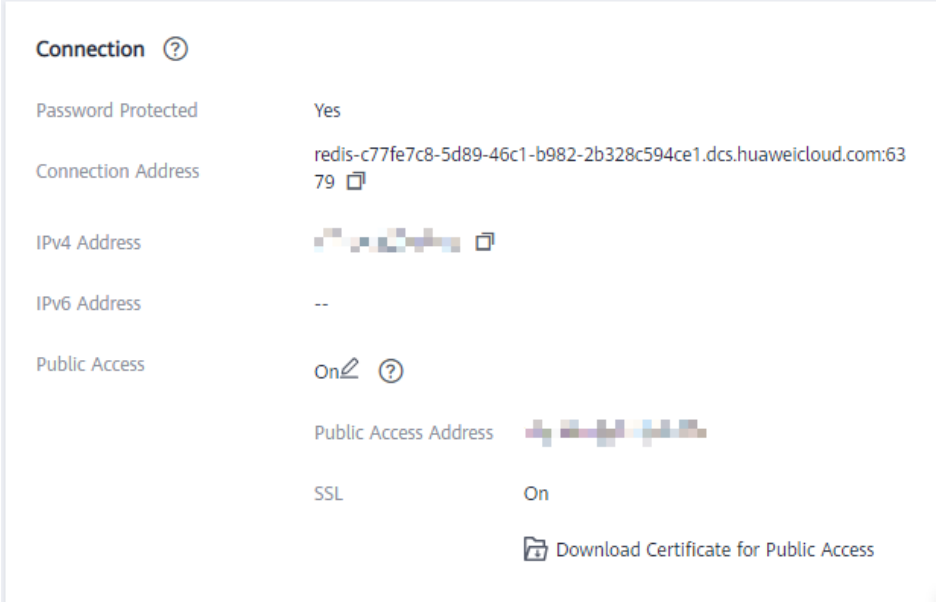
O DCS Redis para 4.0/5.0 suporta apenas transmissão de texto não criptografado. Eles não suportam transmissão criptografada SSL.

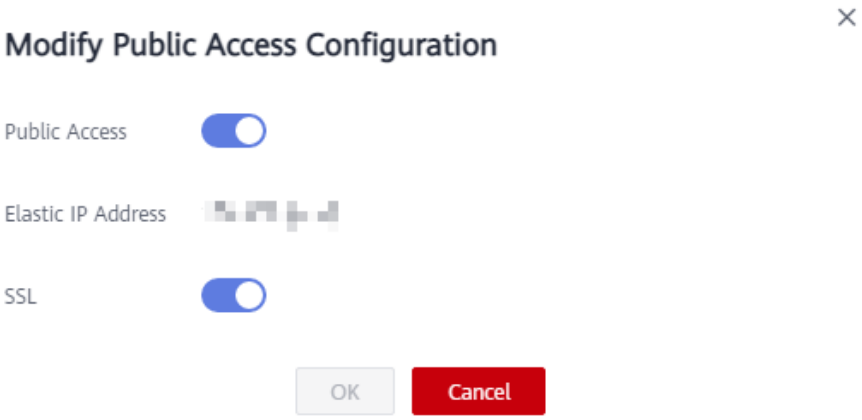
3.25 Como habilitar ou desabilitar o SSL para acesso público a uma instância do DCS Redis 3.0?

Quando você ativa o acesso público, o SSL é habilitado por padrão.

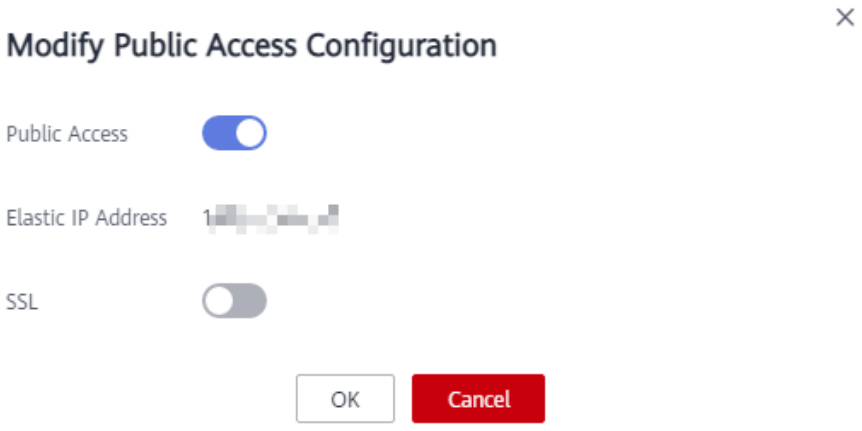
Para desativar a criptografia SSL, execute as seguintes etapas:

1. Abra a página para configurar o acesso público.





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



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

3.26 Por que a memória disponível de instâncias de DCS não usadas é menor que a memória total e por que o uso de memória de instâncias de DCS não usadas é maior que zero?

Para instâncias do DCS Redis 3.0 e instâncias do Memcached, a memória disponível é menor do que a memória total, pois alguma memória é reservada para sobrecarga do sistema e persistência de dados (suportada por instâncias principais/em espera). As instâncias do DCS usam uma certa quantidade de memória para buffers do Redis-server e estruturas de dados internas. É por isso que o uso de memória de instâncias de DCS não utilizadas é maior que zero. Esse problema não ocorrerá em outras versões de instância.

3.27 Como estimar o uso da memória do Redis?

O uso estimado da memória pode ser diferente do uso real da memória. Atualmente, o DCS for Redis fornece as seguintes métricas relacionadas à memória:

Tabela 3-2 Métricas de instância do DCS Redis 3.0

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
memory_usage	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
used_memory	Memória utilizada	Número de bytes usados pelo servidor Redis Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto
used_memory_dataset	Conjunto de dados de memória usado	Memória do conjunto de dados que o servidor Redis utilizou Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Suportado pelo Redis 4.0 e posterior Dimensão: dcs_instance_id	1 minuto
used_memory_dataset_perc	Taxa de conjunto de dados de memória usada	Porcentagem de memória de dados que o Redis usou em relação ao total de memória usada Unidade: %	0–100%	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Suportado pelo Redis 4.0 e posterior Dimensão: dcs_instance_id	1 minuto

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
used_memory_rss	Memória RSS usada	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	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto
memory_frag_ratio	Taxa de fragmentação de memória	Fragmentação de memória atual, que é a proporção entre **used_memory_rss**/ **used_memory**.	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto
used_memory_peak	Pico da memória usada	Memória de pico consumida pelo Redis desde a última inicialização do servidor Redis Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
used_memory_lua	Memória Lua usada	Número de bytes usados pelo mecanismo Lua Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto

Tabela 3-3 Métricas de instância do DCS Redis 4.0 e 5.0

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
memory_usage	Uso da memória	Memória consumida pelo objeto monitorado Unidade: %	0–100%	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
used_memory	Memória utilizada	Número de bytes usados pelo servidor Redis Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto
used_memory_dataset	Conjunto de dados de memória usado	Memória do conjunto de dados que o servidor Redis utilizou Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto
memory_frag_ratio	Taxa de fragmentação de memória	Relação entre memória RSS usada e memória usada	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto
used_memory_lua	Memória Lua usada	Número de bytes usados pelo mecanismo Lua Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
used_memory_peak	Pico da memória usada	Memória de pico consumida pelo Redis desde a última inicialização do servidor Redis Unidade: byte	≥ 0	Objeto monitorado: Instância do DCS Redis de nó único, principal/em espera ou cluster Dimensão: dcs_instance_id	1 minuto

3.28 Por que a capacidade ou o desempenho de uma partição de uma instância de Redis Cluster está sobrecarregado quando a instância ainda está abaixo do gargalo?

O Redis Cluster usa um método especial de fragmentação de dados. **Cada chave faz parte de um slot de hash, que é mantido por um nó no cluster.** Para calcular qual é o slot de hash de uma determinada chave:

1. Pegue o CRC16 da chave no módulo 16384.
2. Com base no mapeamento entre slots de hash e partições, as conexões são redirecionadas para o nó certo para operações de leitura e gravação de dados.

Portanto, as chaves não são distribuídas uniformemente para cada partição de uma instância. Se uma partição contiver uma chave grande ou uma tecla de atalho, a capacidade ou o desempenho da partição será sobrecarregado, mas a carga em outras partições ainda é baixa. Como resultado, o gargalo de capacidade ou desempenho de toda a instância não é atingido.

3.29 O DCS oferece suporte a extensões, plug-ins ou módulos externos?

Não. O DCS for Redis não oferece suporte a extensões, plug-ins ou módulos externos. Não há um plano de suporte aos módulos.

3.30 Por que uma chave desaparece no Redis?

Normalmente, as chaves do Redis não desaparecem. Se uma chave estiver faltando, ela pode ter expirada, sido removida ou excluída.

Execute as seguintes verificações uma a uma:

1. Verifique se a chave expirou.
2. Visualize as informações de monitoramento e verifique se o despejo foi acionado.
3. Execute o comando **INFO** no lado do servidor para verificar se a chave foi excluída.

3.31 Por que ocorre um erro de OOM durante uma conexão do Redis?

Sintoma

"Error in execution; nested exception is io.lettuce.core.RedisCommandExecutionException: OOM command not allowed when used memory > 'maxmemory'" é retornado durante uma conexão de Redis.

Localização de falhas

Um erro de falta de memória (OOM) indica que a memória máxima é excedida. Nas informações de erro, o parâmetro **maxmemory** indica a memória máxima configurada no servidor Redis.

Se o uso de memória da instância do Redis for menor que 100%, a memória do nó onde os dados são gravados pode ter atingido o limite máximo. Conecte-se a cada nó no cluster executando **redis-cli -h <redis_ip> -p 6379 -a <redis_password> -c --bigkeys**. Ao se conectar a um nó de réplica, execute o comando **READONLY** antes de executar o comando **bigkeys**.

3.32 Quais clientes posso usar para o Redis Cluster em diferentes linguagens de programação?

A tabela a seguir compara o Redis Cluster e o Proxy Cluster no DCS.

Tabela 3-4 Comparar o Redis Cluster e o Proxy Cluster

Item	Redis Cluster	Proxy Cluster
Compatibilidade com o Redis	Alto	Médio
Compatibilidade com o cliente	Médio (O modo de cluster deve ser ativado no cliente.)	Alto
Custos	Alto	Médio
Latência	Baixo	Médio
Separação de leitura/gravação	Suporte nativo (configuração do SDK do cliente)	Implementado usando proxies
Desempenho	Alto	Médio

O Redis Cluster não usa proxies e, portanto, oferece menor latência e maior desempenho. No entanto, as instâncias do Redis Cluster são baseadas no protocolo Redis Cluster de código aberto, portanto, sua compatibilidade com o cliente é menor do que a das instâncias do Proxy Cluster.

A tabela a seguir lista os clientes que podem ser usados para o Redis Cluster.

Tabela 3-5 Clientes que podem ser usados para o Redis Cluster

Linguagem	Cliente	Documento de referência
Java	Jedis	**https://github.com/xetorthio/jedis#jedis-cluster**
Java	Lettuce	**https://github.com/lettuce-io/lettuce-core/wiki/Redis-Cluster**
PHP	php redis	**https://github.com/phpredis/phpredis#readme**
Go	Go Redis	Redis Cluster: **https://pkg.go.dev/github.com/go-redis/redis/v8#NewClusterClient** Proxy Cluster, nó único ou principal/em espera: **https://pkg.go.dev/github.com/go-redis/redis/v8#NewClient**
Python	redis-py-cluster	**https://github.com/Grokzen/redis-py-cluster#usage-example**
C	hiredis-vip	**https://github.com/vipshop/hiredis-vip?_ga=2.64990636.268662337.1603553558-977760105.1588733325**
C++	redis-plus-plus	**https://github.com/sewenew/redis-plus-plus?_ga=2.64990636.268662337.1603553558-977760105.1588733325#redis-cluster**
Node.js	node-redis io-redis	**https://github.com/NodeRedis/node-redis** **https://github.com/luin/ioredis**

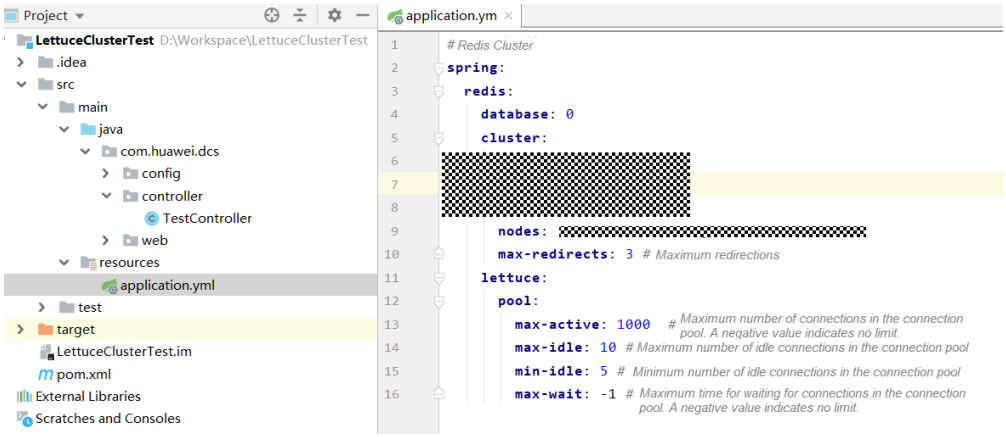
Para exibir todos os clientes do Redis, consulte **https://redis.io/clients**.

3.33 Por que preciso configurar o tempo limite para o Redis Cluster?

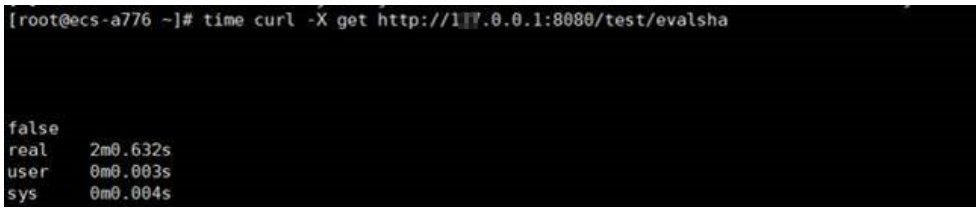
Se o tempo limite não estiver configurado, as conexões falharão.

Quando você se conecta a uma instância do Redis Cluster usando Spring Boot e Lettuce, conecte-se a todos os nós do cluster, incluindo réplicas defeituosas.

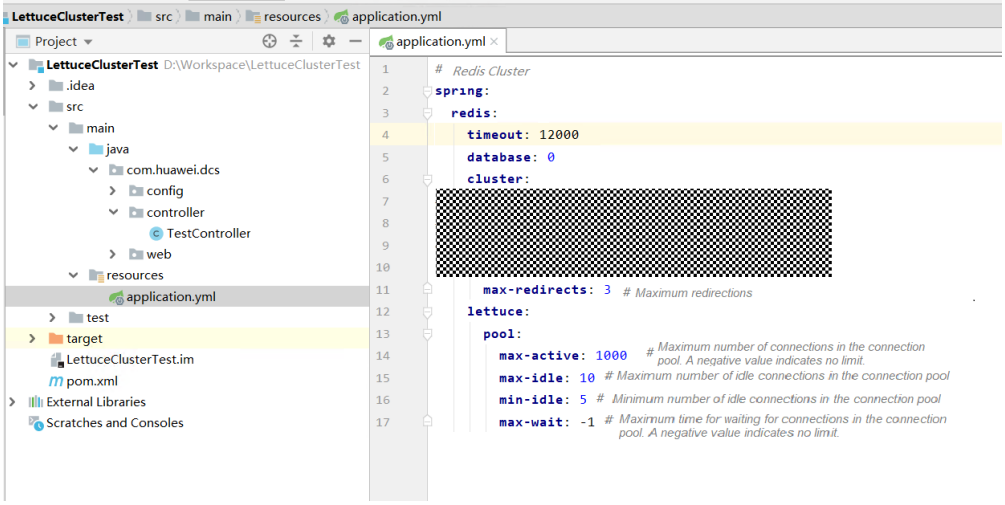
- Se o tempo limite não estiver configurado, o bloqueio em nível de minuto (120s em versões anteriores de Lettuce e 60s na nova versão) pode ocorrer quando há uma réplica defeituosa, como mostrado na figura a seguir.



  O tempo de acesso ao serviço de ponta a ponta pode atingir o tempo limite máximo, conforme mostrado na figura a seguir.



- Depois que o parâmetro **timeout** estiver configurado no cliente, o tempo limite na réplica será muito reduzido. Você pode ajustar o tempo limite com base nos requisitos de serviço. Suponha que a configuração seja a seguinte.



A figura a seguir mostra o tempo de acesso ao serviço de ponta a ponta após a conclusão da configuração.

Se o parâmetro **timeout** não estiver configurado e houver um nó defeituoso, as conexões do cliente serão bloqueadas.

Configure o tempo limite com base no que o serviço pode tolerar. Por exemplo, se você precisar solicitar o Redis duas vezes em uma solicitação HTTP e o tempo limite máximo de uma solicitação HTTP for 10s, é recomendável definir o tempo limite no Redis para 5s. Isso evita a interrupção do serviço se ocorrerem falhas devido a uma longa duração de tempo limite ou nenhuma duração de tempo limite.

3.34 Quais são as restrições na implementação de vários bancos de dados em uma instância de Proxy Cluster?

Observe as seguintes restrições ao considerar a implementação de multi-BD:

- **Restrições de uso:**
  a. O comando **SWAPDB** não suporta multi-BD.
  b. O comando **INFO KEYSPACE** não devolve dados de multi-BD.
  c. Para consultar o número total de chaves em cada banco de dados, use o comando **dbstats** personalizado. O uso da CPU aumentará no nó que estiver executando esse comando.
  d. Os scripts LUA não suportam multi-BD.
  e. O comando **RANDOMKEY** não suporta multi-BD.
  f. O comando **SELECT** não pode ser incorporado em transações.
  g. **PUBLISH** não pode ser usado em scripts Lua.
  h. O número do banco de dados varia de 0 a 255.
  i. As instâncias do DCS Redis 3.0 do Proxy Cluster não oferecem suporte a multi-BD.

- **Restrições de desempenho**

  a. O comando **FLUSHDB** exclui as chaves uma a uma, o que leva muito tempo e é mais lento do que a implementação nativa de código aberto. A velocidade de execução do comando **FLUSHDB** é a mesma do comando **SCAN** (que deve ser testado pelo cliente).

  b. O comando **DBSIZE** é demorado. Não o use no código.

  c. Se multi-BD for usado, o desempenho dos comandos **KEYS** e **SCAN** se deteriora em até 50%.

- **Outras restrições**:

  O armazenamento de back-end reescreve chaves com base em certas regras. As chaves no arquivo RDB exportado não são as chaves originais. No entanto, o acesso através do protocolo Redis não é afetado.

Procedimento para habilitar multi-BD em uma instância de BD único

Por padrão, multi-BD está desabilitado. Antes de ativar ou desativar multi-BD para uma instância, limpe os dados da instância. Faça o seguinte para habilitar multi-BD.

Passo 1 Efetue logon no console do DCS.

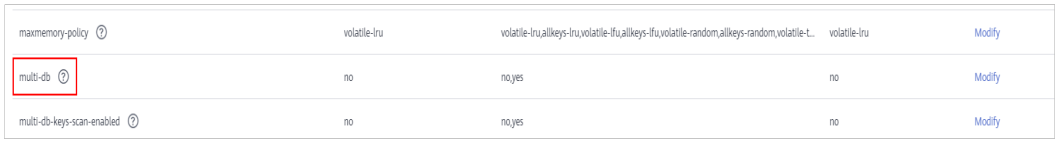
Passo 2 Conecte-se à instância e execute o comando **FLUSHALL** para limpar os dados da instância.

Passo 3 Na página **Cache Manager** do console do DCS, clique na instância de DCS desejada.

Passo 4 Escolha **Instance Configuration** > **Parameters**.

Passo 5 Clique em **Modify** na linha que contém o parâmetro **multi-db** e altere o respectivo valor para **yes**.

Passo 6 Clique em **Save** e confirme a modificação. A instância não precisa ser reiniciada.



----Fim

3.35 Posso alterar a AZ de uma instância?

Não.

Se quiser usar uma AZ diferente, crie outra instância na AZ desejada e migre os dados.

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

- A instância de destino está disponível. Se você já tiver uma instância do DCS Redis, use-a diretamente e limpe os dados da instância antes da migração. Para obter detalhes, consulte **Limpeza de dados de instância de DCS**.

  Se os dados da instância de destino não forem limpos 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.
- Os recursos do Redis de destino, do Redis de origem e da tarefa de migração estão na mesma VPC.

  NOTA

  Se as instâncias do Redis de destino e de origem não estiverem na mesma VPC, verifique se os recursos de VM da tarefa de migração podem acessar essas instâncias.

  - Se as instâncias de Redis de origem e de destino estiverem na mesma região, crie uma conexão de emparelhamento VPC consultando **Conexão de emparelhamento de VPC**.
  - Se as instâncias de Redis de origem e de destino estiverem em regiões diferentes, crie uma conexão de nuvem consultando **Primeiros passos da Cloud Connect**.
- As instâncias de destino e de origem usam a mesma porta.
- A comutação de 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 obter detalhes, consulte **Modos de migração de dados de DCS**.
  - A tabela seguinte lista os cenários de comutação IP suportados.

Tabela 3-6 Cenários de comutação IP

Fonte	Alvo
Nó único, divisão de leitura/gravação ou principal/em espera	Nó único, divisão de leitura/gravação, principal/em espera ou Proxy Cluster
Proxy Cluster	Nó único, divisão de leitura/gravação, principal/em espera ou Proxy Cluster

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

3.36 Explicação e uso de hashtags

Design de hashtag

Operações multi-chave, como aquelas que usam o comando **MSET** ou scripts Lua, são atômicas. Todas as chaves especificadas são executadas ao mesmo tempo. No entanto, em um cluster, cada chave é hash para uma determinada partição e as operações de várias chaves não são mais atômicas. As chaves podem ser alocadas para diferentes slots. Como resultado, algumas chaves são atualizadas, enquanto outras não. Se houver uma hashtag, o cluster determinará qual slot alocar uma chave com base na hashtag. Chaves com a mesma hashtag são alocadas para o mesmo slot.

Uso de hashtags

Somente o conteúdo entre a primeira chave à esquerda ({) e a primeira chave à direita (}) a seguir é submetido a hash.

Por exemplo:

- Nas teclas **{user1000}.following** e **{user1000}.followers**, há apenas um par de chaves. **user1000** será hash.
- Na tecla **foo{}{bar}**, não há conteúdo entre a primeira { e a primeira }. Toda a tecla **foo{}{bar}** será hash como de costume.
- Na tecla **foo{{bar}}zap**, **{bar** (o conteúdo entre a primeira { e a primeira }) é hash.
- Na tecla **foo{bar}{zap}**, **bar** é hash porque está entre o primeiro par de { e }.

Exemplo de hashtag

Quando a seguinte operação é realizada:

```
EVAL "redis.call('set',KEYS[1],ARGV[1]) redis.call('set',KEYS[2],ARGV[2])" 2 key1
key2 value1 value2
```

O seguinte erro é exibido:

ERR 'key1' e 'key2' não estão no mesmo slot

Você pode usar uma hashtag para resolver esse problema:

```
EVAL "redis.call('set',KEYS[1],ARGV[1]) redis.call('set',KEYS[2],ARGV[2])" 2
{user}key1 {user}key2 value1 value2
```

3.37 Os dados armazenados em cache serão retidos após uma instância ser reiniciada?

Depois que uma instância de DCS de nó único é reiniciada, os dados na instância são excluídos.

As instâncias principal/em espera e de cluster (exceto clusters de réplica única) suportam a persistência AOF por padrão. Os dados são retidos depois que essas instâncias são reiniciadas.

Se a persistência AOF estiver desabilitada (**appendonly** está definido como **no**), os dados serão excluídos depois que as instâncias forem reiniciadas.

3.38 Como comprar uma instância de Proxy Cluster de vários bancos de dados?

Quando você compra uma instância de Proxy Cluster, há apenas um banco de dados por padrão. Esta seção descreve como comprar uma instância de Proxy Cluster com vários bancos de dados.

NOTA

Antes de começar, aprenda sobre **as restrições na implementação de multi-BD**.

Passo 1 Faça logon no **console de DCS**.

Passo 2 Clique em no canto superior esquerdo para selecionar uma região.

Passo 3 No painel de navegação, escolha **Parameter Templates**.

Passo 4 Na linha que contém o modelo com a versão do mecanismo de cache desejada e o tipo de instância (Proxy Cluster), clique em **Customize**.

Passo 5 Defina **multi-db** como **yes**.

Passo 6 Insira um novo nome de modelo e clique em **OK**. O modelo personalizado foi criado com sucesso.

Passo 7 No painel de navegação, escolha **Cache Manager**. Em seguida, clique em **Buy DCS Instance** para criar uma instância de Proxy Cluster.

Defina **Parameter Configuration** para **Use custom template** e selecione o modelo personalizado criado na etapa anterior.

Depois que a instância for criada, conecte-se a ela para verificar se ela tem vários bancos de dados.

----Fim

3.39 Por que uma instância é congelada?

As instâncias estão no estado **Frozen** se o pacote anual/mensal não for renovado após a expiração. As instâncias congeladas ainda estão em execução, mas não podem ser usadas até que você renove o pacote.

Depois que uma instância é congelada, há um período de retenção de cinco dias. Se o pacote não for renovado dentro desse período, a instância será excluída.

4 Dimensionamento e atualização de instância

4.1 Posso atualizar a versão de uma instância de DCS Redis, por exemplo, do Redis 4.0 para o Redis 5.0?

Não. Diferentes versões do Redis usam diferentes arquiteturas subjacentes. A versão do Redis usada por uma instância do DCS não pode ser alterada depois que a instância é criada.

Se o seu serviço exigir os recursos de versões superiores do Redis, crie uma nova instância de DCS Redis de uma versão superior e migre os dados da instância original para a nova. Para obter detalhes sobre como migrar dados, consulte o **Guia de migração de dados**.

4.2 Os serviços são interrompidos se a manutenção for executada durante a janela de tempo de manutenção?

O pessoal de O&M entrará em contato com você antes de realizar a manutenção durante a janela de tempo de manutenção, informando-o sobre as operações e seus impactos. Você não precisa se preocupar com exceções em execução de instância.

4.3 As instâncias são interrompidas ou reiniciadas durante a modificação da especificação?

Não. As modificações de especificação podem ocorrer enquanto a instância está em execução e não afetam outros recursos.

4.4 Quais alterações de tipo de instância do DCS são suportadas?

Tabela 4-1 Opções de alteração do tipo de instância suportadas por diferentes instâncias de DCS

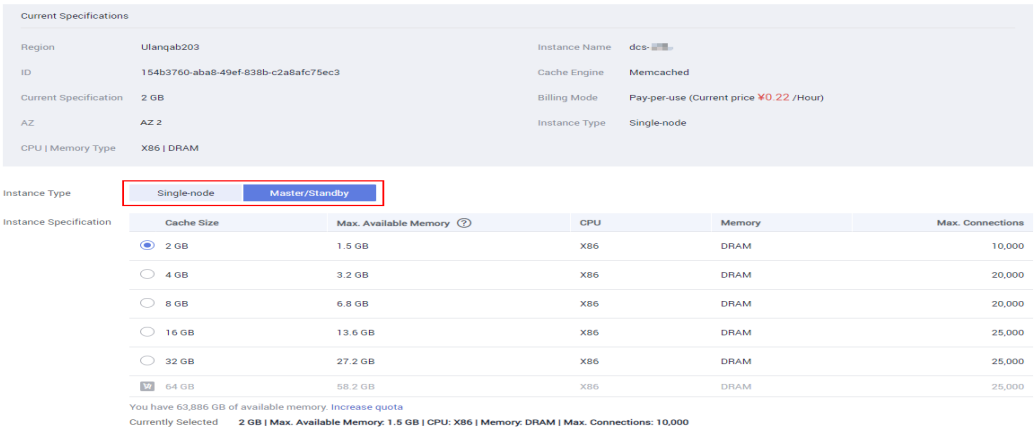
Versão	Alteração de tipo suportada	Precauções
Redis 3.0	De nó único para principal/em espera	A instância não pode ser conectada por vários segundos e permanece somente leitura por cerca de um minuto.
	Do principal/em espera para o cluster de proxy	1. Se os dados de uma instância principal/em espera do 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.

Para obter detalhes sobre os comandos suportados por diferentes tipos de instâncias, consulte **Compatibilidade de comandos**.

Quaisquer alterações de tipo de instância não listadas na tabela anterior não são suportadas. Para modificar as especificações ao alterar o tipo de instância, consulte **Comutação de IP**.

Para verificar se é possível alterar o tipo de instância de uma instância, consulte os parâmetros exibidos na página **Modify Specifications** no console do DCS. O cenário a seguir mostra que o tipo de instância pode ser alterado.



4.5 Os serviços são interrompidos durante a modificação da especificação?

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.

A tabela a seguir descreve o impacto da modificação da especificação.

Alteração do tipo de instância

Tabela 4-2 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.

Versão	Alteração de tipo suportada	Precauções
Memcached	De nó único para principal/em espera	Os serviços são interrompidos por vários segundos e permanecem somente leitura por cerca de 1 minuto.
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 4-3 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	-
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 4-4 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.

4.6 Por que não modificar as especificações de uma instância de DCS?

- Verifique se outras tarefas estão em execução.

  As especificações de uma instância de DCS não podem ser modificadas se outra tarefa da instância ainda estiver em execução. Por exemplo, você não pode excluir ou ampliar uma instância enquanto ela está sendo reiniciada. Da mesma forma, você não pode excluir uma instância enquanto ela está sendo ampliada.

  Se a modificação da especificação falhar, tente novamente mais tarde. Se falhar novamente, entre em contato com o suporte técnico.
- Ao alterar uma instância principal/em espera para o tipo de Proxy Cluster, verifique se existem dados em bancos de dados diferentes de DB0. A modificação da especificação falhará se um banco de dados diferente do DB0 contiver dados.

  Uma instância principal/em espera pode ser alterada para o tipo de Proxy Cluster quando os dados existem apenas no DB0.

4.7 Como reduzir a capacidade de uma instância de DCS?

Tabela 4-5 lista opções de dimensionamento suportadas por diferentes instâncias de DCS.

Tabela 4-5 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	-
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.	-	-	-

Para obter detalhes sobre como alterar a capacidade, consulte **Modificação de especificações**.

Se você quiser usar uma instância menor do Proxy Cluster de DCS Redis 3.0, faça o backup dos dados da instância existente e crie uma nova instância do Proxy Cluster com a capacidade desejada. Em seguida, importe os dados de backup para a nova instância. Após a conclusão da migração de dados, exclua a instância anterior. Para obter detalhes sobre as operações de migração de dados, consulte **Importação de arquivos de backup**.

4.8 Como adicionar partições a uma instância do DCS Redis de cluster sem alterar a memória?

Depois que uma instância de Proxy Cluster ou Redis Cluster é criada, você pode reduzir a capacidade de cada partição e adicionar mais partições sem alterar a memória total.

Por exemplo, se uma instância de 8 GB tiver 4 partições e cada partição tiver 2 GB, você poderá reduzir o tamanho da partição para 1 GB e aumentar a quantidade de partições para 8.

> **NOTA**
>
> Um tamanho de partição de 1 GB não pode ser alterado.

Procedimento

Passo 1 Faça logon no **console do DCS**.

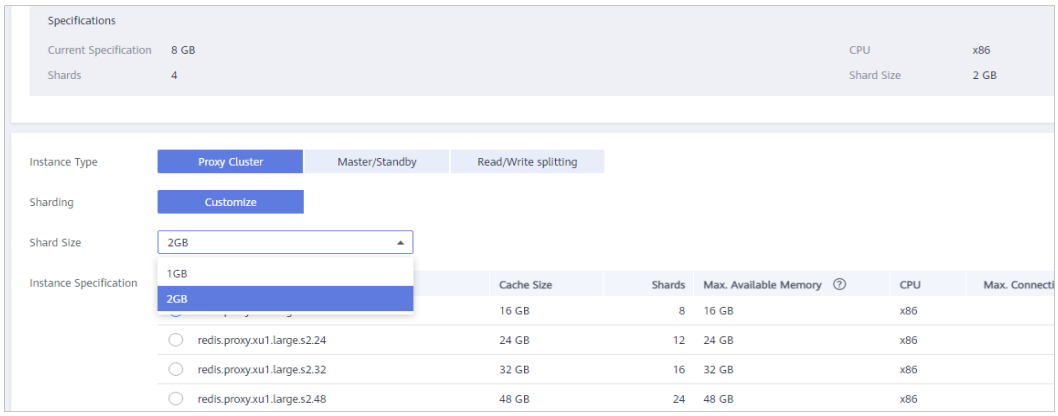
Passo 2 Clique em no canto superior esquerdo para selecionar uma região e um projeto.

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

Passo 4 Escolha **More** > **Modify Specifications** na linha que contém a instância de DCS desejada.

Passo 5 Na página **Modify Instance Specifications** exibida, especifique **Shard Size** e **Instance Specification**.

Figura 4-1 Selecionar um tamanho de partição



Passo 6 Clique em **Next**, confirme os detalhes e clique em **Submit**.

A modificação leva cerca de 5 a 30 minutos para ser concluída. Depois que a modificação é bem-sucedida, o status da instância muda para **Running**.

----Fim

4.9 Como lidar com um erro quando uso Lettuce para conectar-se a uma instância de Redis Cluster após a modificação da especificação?

Sintoma

Se a quantidade de partições for alterada durante a modificação da especificação de uma instância do Redis Cluster, alguns slots serão migrados para novas partições. O erro a seguir ocorre quando você usa Lettuce para se conectar à instância.

Figura 4-2 Erro

```
org.springframework.data.redis.RedisSystemException: Redis exception; nested exception is io.lettuce.core.RedisException: io.lettuce.core.R
edisException: java.lang.IllegalArgumentException: Connection to [redacted] not allowed. This connection point is not known in the
 cluster view
```


Para obter detalhes, consulte **Conexão com X não permitida. Este ponto de conexão não é conhecido na vista de cluster**.

Análise

Processo de modificação de especificação de uma instância do Redis Cluster:

Após ser iniciado, o cliente obtém a topologia do nó de cluster usando o comando **Cluster Nodes** baseado em RESP2 e mantém a topologia em sua estrutura de dados na memória.

Para acesso a dados, o cliente usa o algoritmo CRC16 para calcular o slot de hash de uma chave e encaminha automaticamente as solicitações com base na topologia e nas informações de slot armazenadas na memória.

Se o número de partições mudar durante o dimensionamento, a topologia e o mapeamento de slot serão alterados. Nesse caso, o cliente precisa atualizar automaticamente a topologia. Caso contrário, a rota de solicitação pode falhar ou o local da rota pode estar incorreto. Como resultado, um erro é relatado durante a conexão do cliente.

Por exemplo, quando o número de partições em uma instância do Redis Cluster muda de três para seis, a topologia e o mapeamento de slots são alterados, conforme mostrado nas figuras a seguir.

Figura 4-3 Uma instância do Redis Cluster antes da expansão

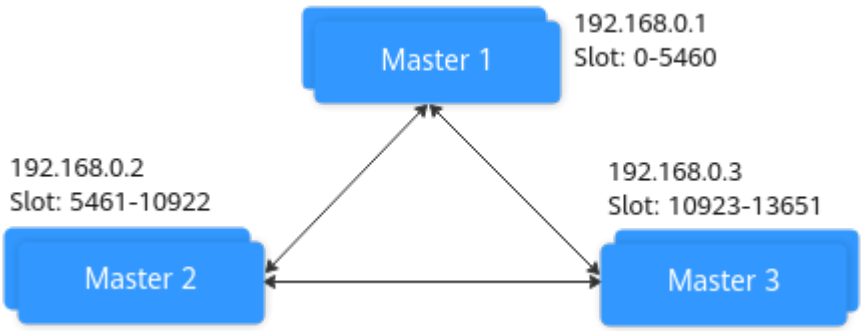
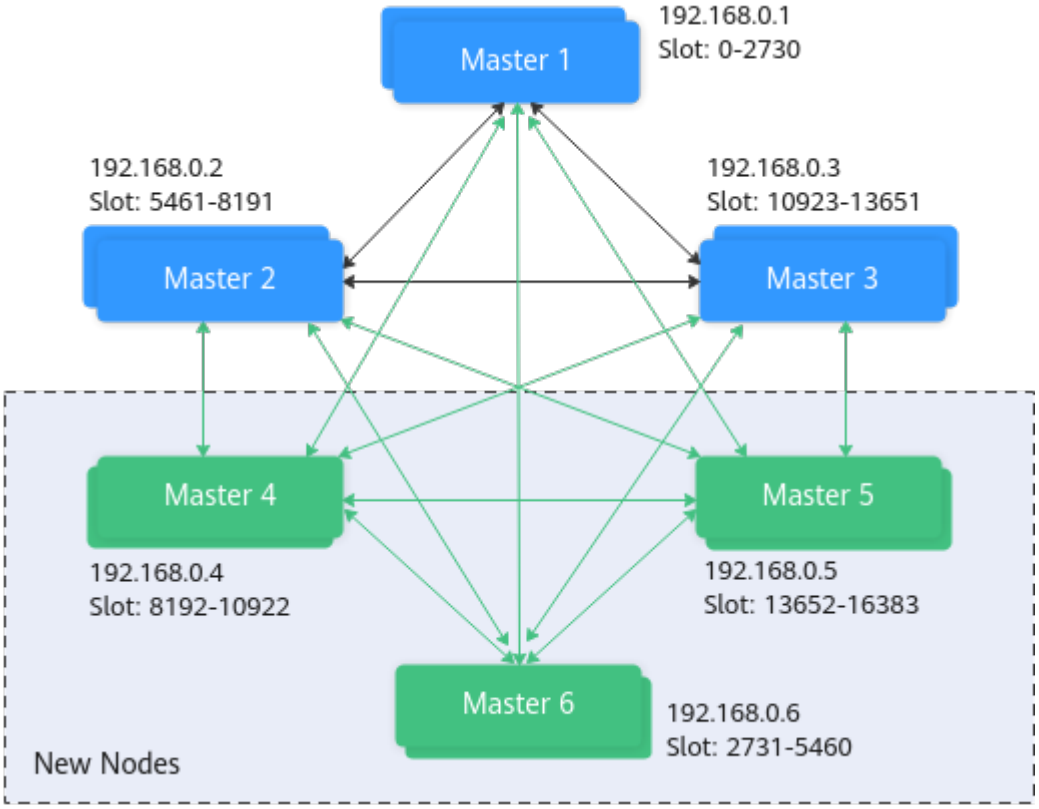


Figura 4-4 Uma instância do Redis Cluster após a expansão



Soluções

Solução 1 (recomendada)

Habilitar a atualização automatizada da topologia.

```
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();
```

Para obter detalhes, consulte **um exemplo de uso do Lettuce para se conectar a uma instância de Redis Cluster**.

NOTA

Se você usar o Lettuce para se conectar a uma instância do Redis Cluster e a atualização automatizada não estiver ativada, será necessário reiniciar o cliente após a modificação da especificação.

Solução 2

Desabilitar a validação da associação ao nó do cluster.

```
ClusterClientOptions clusterClientOptions = ClusterClientOptions.builder()
        .validateClusterNodeMembership(false)
        .build();
```


Se **validateClusterNodeMembership** for **true**, verifique se o endereço de conexão atual está na topologia do cluster obtida por meio de **CLUSTER NODES**, antes de se conectar ao cluster. Se não estiver na topologia, o erro ocorre.

NOTA

Impacto da desativação da validação da associação ao nó do cluster:

- Falta de detecção de violação de segurança.
- Se a atualização automática da topologia estiver desativada, uma solicitação de redirecionamento MOVED poderá ser gerada depois que as especificações do Redis Cluster forem alteradas e a quantidade de partições aumentar. O redirecionamento aumenta a carga de rede do cluster e o tempo necessário para processar uma única solicitação. Se a quantidade de partições diminuir, as partições excluídas não poderão ser conectadas.

4.10 Posso expandir uma partição única de uma instância de cluster?

Não. Você só pode adicionar mais partições para expandir a capacidade da instância.

Se você quiser usar um tamanho maior em cada partição de uma instância de Proxy Cluster, altere a instância para o tipo principal/em espera e, em seguida, altere-a de volta para Proxy Cluster com o tamanho de partição desejada. Antes de alterar o tipo de instância para Proxy Cluster, avalie o impacto nos serviços. Para mais detalhes, consulte **Quais são as restrições na implementação de vários bancos de dados em uma instância de Proxy Cluster?**.

5 Backup, exportação e migração de dados

5.1 Como exportar dados de instância do DCS Redis?

- Principal/em espera, divisão de leitura/gravação e instâncias de cluster:

  Essas instâncias suportam backups. Execute as seguintes operações para exportar dados:

  a. Na página **Backups and Restorations**, exiba as tarefas de backup.

  b. Se não houver backup, crie um backup e baixe o arquivo de backup conforme solicitado.

  NOTA

  Se suas instâncias de DCS foram criadas há muito tempo, as versões dessas instâncias podem não estar avançadas o suficiente para suportar algumas novas funções (como backup e restauração). Você pode entrar em contato com o suporte técnico para atualizar suas instâncias de DCS. Após a atualização, você pode fazer backup e restaurar suas instâncias.

- Instâncias de nó único:

  As instâncias de nó único não suportam a função de backup. Você pode usar o redis-cli para exportar dados para arquivos RDB. Esta operação depende do comando **SYNC**.

  – Se a instância permitir o comando **SYNC** (como uma instância de nó único do Redis 3.0), execute o seguinte comando para exportar os dados da instância:

    redis-cli -h {source_redis_address} -p 6379 [-a password] --rdb {output.rdb}

  – Se a instância não permitir o comando **SYNC** (como uma instância de nó único do Redis 4.0 ou 5.0), migre os dados da instância para uma instância principal/em espera e exporte os dados usando a função de backup.

5.2 Por que a memória de uma instância do DCS Redis não é alterada após a migração de dados usando Rump, mesmo que nenhuma mensagem de erro seja retornada?

Para obter detalhes sobre como usar o Rump, consulte o **Guia de migração de dados**.

Possíveis causas:

- O Rump não oferece suporte à migração para instâncias de DCS de cluster.
- Os comandos são executados incorretamente no Rump.

5.3 Posso exportar dados de backup de instâncias do DCS Redis para arquivos RDB no console?

- Instâncias de DCS Redis 3.0

  Não. No console, os dados de backup de uma instância do DCS Redis 3.0 podem ser exportados somente para arquivos AOF. Para exportar dados para arquivos RDB, execute o seguinte comando no redis-cli:

  redis-cli -h *{redis_address}* -p 6379 [-a *password*] --rdb {output.rdb}

- Instâncias do DCS Redis 4.0 e 5.0

  Sim. As instâncias do DCS Redis 4.0 e 5.0 são compatíveis com a persistência de AOF e RDB. Você pode fazer backup de dados em arquivos RDB e AOF no console e fazer o download dos arquivos.

5.4 Por que os processos são interrompidos com frequência durante a migração de dados?

Possível causa: a memória é insuficiente.

Solução: expanda a memória do servidor no qual o comando de migração é executado.

5.5 Onde os arquivos de backup da instância de DCS são armazenados? Como são cobrados?

Os arquivos de backup são armazenados no OBS. Atualmente, o DCS e o OBS não cobram por backups. No futuro, uma certa quantidade de taxas pode ser cobrada com base no padrão unificado.

5.6 Todos os dados em uma instância do DCS Redis são migrados durante a migração on-line?

A migração entre instâncias de nó único e principal/em espera envolve o conjunto completo de dados. Todos os BDs serão migrados e você não poderá migrar BDs especificados. Após a migração, uma determinada chave permanecerá no mesmo banco de dados de antes da migração.

Por outro lado, uma instância de cluster tem apenas um banco de dados, que é o DB0. Durante a migração, os dados em todos os slots do DB0 são migrados.

5.7 O DCS suporta a persistência de dados? Qual é o impacto da persistência?

Apoio à persistência

- Instâncias do DCS Redis:
  - Nó único: a persistência de dados não é suportada.
  - Principal/em espera, divisão de leitura/gravação e cluster (exceto clusters de réplica única): a persistência de dados é suportada.
- Instâncias de DCS Memcached:
  - Nó único: a persistência de dados não é suportada.
  - Principal/em espera: a persistência de dados é suportada.

Modos de persistência

- O DCS suporta apenas a persistência AOF por padrão. Você pode ativar ou desativar a persistência conforme necessário. Todas as instâncias, exceto as de cluster de nó único e de réplica única, são criadas com persistência AOF habilitada.
- Os DCS não oferecem suporte à persistência RDB por padrão e você não pode configurar o parâmetro **save**. Se a persistência RDB for necessária para uma instância principal/em espera ou de Redis Cluster 4.0 ou posterior, você poderá usar a função de backup e restauração para fazer backup dos dados da instância em um arquivo RDB e armazenar os dados no OBS.

Disco usado para persistência

Para instâncias do DCS Redis 4.0 e posteriores, os dados são mantidos em discos SSD.

Impacto da persistência AOF

Depois que a persistência AOF é ativada, o processo de Redis-Server precisa registrar operações no arquivo AOF para persistência de dados.

- Se o disco ou I/O do nó de computação subjacente estiver com defeito, a latência pode aumentar ou pode ocorrer uma alternância principal/em espera.
- O Redis-Server reescreve periodicamente AOF. Durante uma reescrita, a latência pode ser alta por um curto período de tempo. Para obter detalhes sobre as regras de reescrita de AOF, consulte **Quando as reescritas de AOF serão acionadas?**

Se as instâncias do DCS forem usadas para acelerar aplicações, é recomendável desativar a persistência para obter maior desempenho e estabilidade. Tenha cuidado ao desabilitar a persistência. Sem persistência, os dados em cache podem ser perdidos em cenários extremos (por exemplo, quando os nós principais e em espera estão com defeito).

Para desabilitar a persistência AOF, defina o parâmetro **appendonly** como **no** na página de detalhes da instância.

Posso ativar a persistência apenas em réplicas e não em principais?

Sim. Para uma instância básica do DCS Redis 4.0/5.0/6.0 principal/em espera ou de cluster, você pode definir o parâmetro **appendonly** como **only-replica** para habilitar a persistência somente em réplicas.

Esta função não está disponível para outras versões ou tipos de instância.

NOTA

- Por padrão, o parâmetro **appendonly** tem apenas duas opções: **yes** e **no**. Para defini-lo como **only-replica**, entre em contato com o pessoal de O&M.
- Sem escrever e reescrever AOF no nó principal, a persistência apenas em réplicas melhora o desempenho, mas reduz a confiabilidade, em comparação com a persistência em nós principais e de réplica. Selecione um modo com base nos requisitos de serviço.

5.8 Quando as reescritas de AOF serão acionadas?

As reescritas de AOF envolvem os seguintes conceitos:

- Janela de reescrita, que atualmente é de 01:00 a 04:59
- Limite de uso do disco, que é de 50%

As reescritas de AOF são acionadas nos seguintes cenários:

- Se o uso do disco atingir o limite (independentemente de a hora atual estar dentro da janela de reescrita) reescritas serão acionadas em instâncias cujo tamanho do arquivo AOF é maior que o tamanho do conjunto de dados de memória.
- Se o uso do disco estiver abaixo do limite e o tempo atual estiver dentro da janela de reescrita, as reescritas serão acionadas em instâncias cujo tamanho do arquivo AOF seja maior que a memória do conjunto de dados multiplicada por 1,5.
- Se o uso do disco estiver abaixo do limite, mas o tempo atual estiver fora da janela de reescrita, as reescritas serão acionadas em instâncias cujo tamanho de arquivo AOF seja maior que a memória máxima multiplicada por 4,5.

5.9 Quais são as causas comuns das falhas de migração do Redis?

- Verifique se ocorreu uma alternância principal/em espera durante a migração. Se ocorrer, entre em contato com o suporte técnico para desativar temporariamente a alternância principal/em espera até que a migração seja concluída.
- Para a migração on-line, verifique se os comandos **SYNC** e **PSYNC** estão desabilitados na instância do Redis de origem. Se eles estiverem desativados, ative-os para permitir a sincronização de dados.
- Por padrão, uma instância do Proxy Cluster possui apenas um banco de dados (DB0). Antes de migrar dados de uma instância de nó único ou principal/em espera para uma instância de Proxy Cluster, verifique se existem dados em bancos de dados diferentes do DB0. Se sim, ative multi-BD para a instância de Proxy Cluster consultando **Ativação de multi-BD**.
- Por padrão, uma instância de Redis Cluster tem apenas um BD (DB0). Antes de migrar dados de uma instância de nó único ou principal/em espera para uma instância de Redis

Cluster, 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 on-line com rump**.

5.10 Posso migrar dados para várias instâncias de destino em uma tarefa de migração?

Não. Uma tarefa de migração permite que os dados sejam migrados para apenas uma instância de destino. Para migrar dados para várias instâncias de destino, crie várias tarefas de migração.

5.11 Como habilitar os comandos SYNC e PSYNC?

- Migração dentro do DCS:
  - Por padrão, os comandos **SYNC** e **PSYNC** podem ser usados quando o Redis auto-hospedado é migrado para o DCS.
  - Durante a migração on-line entre instâncias do DCS Redis na mesma região sob a mesma conta, os comandos **SYNC** e **PSYNC** são ativados automaticamente.
  - Durante a migração on-line entre instâncias do DCS Redis em regiões diferentes ou em contas diferentes em uma região, os comandos **SYNC** e **PSYNC** não são ativados automaticamente e a migração on-line não pode ser usada. Você pode migrar dados usando arquivos de backup.
- Migração de outros fornecedores de nuvem para o DCS:
  - Geralmente, os fornecedores de nuvem desabilitam os comandos **SYNC** e **PSYNC**. Se você quiser usar a função de migração on-line no console do DCS, entre em contato com a equipe de O&M do fornecedor de nuvem de origem para ativar os comandos. Para a migração off-line, você pode importar arquivos de backup.
  - Se a migração incremental não for necessária, você poderá realizar a migração completa consultando **Migração completa on-line do Redis de outra nuvem com redis-shake**. Este método não depende de **SYNC** e **PSYNC**.

5.12 As mesmas chaves serão substituídas durante a migração de dados ou a importação de backup?

Se os dados existirem nas instâncias de origem e de destino, os dados de destino serão sobrescritos pelos dados de origem. Se os dados existirem apenas na instância de destino, os dados serão retidos.

A inconsistência entre os dados de origem e de destino após a migração pode ser devido aos dados de destino que existiam e foram retidos antes da migração.

6 Análise de tecla grande, análise de tecla de atalho e varredura de chave expirada

6.1 Como analisar as teclas de atalho de uma instância do DCS Redis 3.0?

O DCS for Redis 3.0 não oferece suporte à análise de teclas de atalho no console. Como alternativa, você pode usar os seguintes métodos para analisar teclas de atalho:

- Método 1: analise a estrutura do serviço e a implementação do serviço para descobrir possíveis teclas de atalho.

  Por exemplo, teclas de atalho podem ser facilmente encontradas no código de serviço durante as vendas flash ou logons de usuário.

  Vantagens: simples e fácil de implementar.

  Desvantagens: requer familiaridade com o código de serviço. Além disso, a análise se torna mais difícil à medida que os cenários de serviço se tornam mais complexos.
- Método 2: colete estatísticas de acesso à chave no código do cliente para descobrir as teclas de atalho.

  Desvantagens: requer modificação de código intrusiva.
- Método 3: capture e analise pacotes.

  Vantagens: simples e fácil de implementar.

6.2 Como o DCS exclui chaves expiradas?

Pergunta

Quais são as regras para a exclusão programada de chaves expiradas diariamente? Posso personalizar as regras?

Mecanismos para excluir chaves expiradas

- Exclusão inerte: a estratégia de exclusão é controlada no loop de eventos de I/O 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 expirada, ela será excluída e uma resposta será retornada indicando que a chave não existe. Se a chave não tiver expirada, 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.

Soluções

- Configure tarefas agendadas de análise de teclas de atalho consultando **Análise de teclas de atalho** ou use o comando **SCAN** para percorrer todas as chaves em uma base agendada e remover chaves expiradas da memória.
- Configure uma tarefa agendada para verificar todos os nós principais da instância. Todas as chaves serão verificadas e o Redis determinará se as chaves expiraram. Chaves expiradas serão liberadas. Para obter detalhes, consulte **Verificação de chaves expiradas**.

6.3 Por quanto tempo as chaves são armazenadas? Como configurar a expiração da chave?

- Duração do armazenamento da chave
  - As chaves que não têm uma expiração são armazenadas permanentemente.
  - As chaves que têm uma expiração são excluídas após expirarem. Para obter detalhes, consulte **Verificação de chaves expiradas**.
  - Para remover a configuração expirada de uma chave, execute o comando **PERSIST**.
- Configurar a expiração da chave

  Você pode executar o comando **EXPIRE** ou **PEXPIRE** para definir o tempo de expiração da chave. Por exemplo, se você executar **expire key1 100**, key1 expirará em 100 segundos. Se você executar **pexpire key2 1800**, key2 expirará em 1800 milissegundos.

  EXPIRE define a expiração da chave em segundos e **PEXPIRE** define a expiração da chave em milissegundos.

6.4 Por que o uso da memória diminui depois que a análise de teclas grandes é executada no Redis?

A análise de teclas grandes apenas consulta teclas que ocupam um grande espaço e não exclui as teclas. Devido ao mecanismo de exclusão lazy free, as teclas não são excluídas imediatamente após a expiração, a menos que sejam acessadas ou identificadas. Durante uma análise de tecla grande, todas as teclas são percorridas e as teclas expiradas serão identificadas

e, em seguida, excluídas. Como resultado, o uso de memória diminui após a conclusão de uma análise de tecla grande.

Para obter detalhes sobre como teclas expiradas são excluídas e como procurar teclas expiradas, consulte **Verificação de teclas expiradas**.

7 Comandos do Redis

7.1 Como limpar dados do Redis?

Tenha cuidado ao limpar dados.

- Redis 3.0

  Os dados de uma instância do DCS Redis 3.0 não podem ser apagados no console e só podem ser apagados pelo comando **FLUSHDB** ou **FLUSHALL** no redis-cli.

  Execute o comando **FLUSHALL** para limpar todos os dados na instância.

  Execute o comando **FLUSHDB** para limpar os dados no banco de dados atualmente selecionado.

- Redis 4.0 ou mais recente

  Para limpar dados de uma instância do DCS Redis 4.0 ou posterior, você pode executar o comando **FLUSHDB** ou **FLUSHALL** no redis-cli, usar a função de limpeza de dados no console do DCS ou executar o comando **FLUSHDB** na CLI da Web.

  Para limpar dados de uma instância de Redis Cluster, execute o comando **FLUSHDB** ou **FLUSHALL** em cada fragmento da instância. Caso contrário, os dados podem não ser completamente limpos.

  NOTA

  - Atualmente, a função de limpeza de dados e o acesso à CLI da Web no console são suportados apenas por instâncias do DCS Redis 4.0 ou posteriores.
  - Quando você executa o comando **FLUSHDB** na CLI da Web, somente uma partição é limpo por vez. Se houver várias partições, conecte-se ao nó principal de cada partição e execute o comando **FLUSHDB** separadamente.
  - Os dados do Redis Cluster não podem ser apagados usando a CLI da Web.

7.2 Como encontrar chaves especificadas e percorrer todas as chaves?

Localizar chaves especificadas

A análise de teclas grandes e teclas de atalho não oferece suporte à pesquisa de teclas com condições especificadas. Para encontrar chaves com o prefixo ou sufixo especificado, use o comando **SCAN**.

Por exemplo, para procurar chaves que contenham a letra *a* em uma instância do Redis, execute o seguinte comando no redis-cli:

./redis-cli -h *{redis_address}* -p *{port}* [-a *password*] --scan --pattern '*a*'

Percorrer todas as chaves

Não utilize o comando **KEYS** para percorrer todas as chaves de uma instância porque o comando **KEYS** é complexo e pode bloquear o Redis. Para percorrer todas as chaves em uma instância do DCS Redis, execute o seguinte comando no redis-cli:

./redis-cli -h *{redis_address}* -p *{port}* [-a *password*] --scan --pattern '*'

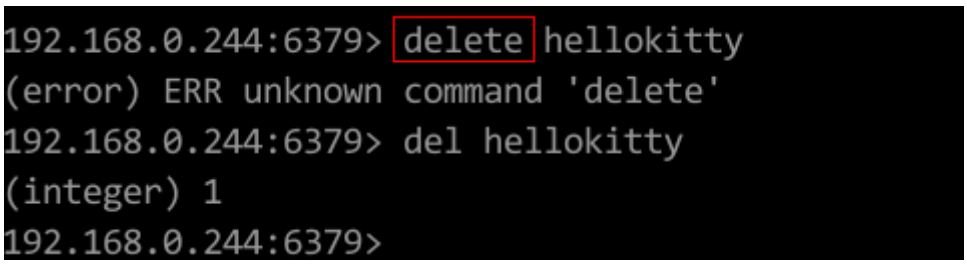
Para obter detalhes sobre o comando **SCAN**, visite o **site oficial do Redis**.

7.3 Por que não consigo executar alguns comandos do Redis?

Possíveis causas incluem o seguinte:

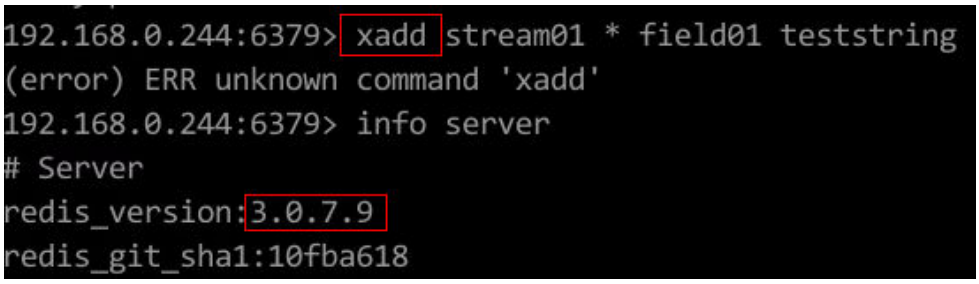
- O comando está escrito incorretamente.

  Como mostrado na figura a seguir, a mensagem de erro é retornada porque o comando correto para excluir uma chave deve ser **del**.

- Um comando disponível em uma versão mais alta do Redis é executado em uma versão mais baixa do Redis.

  Conforme mostrado na figura a seguir, a mensagem de erro é retornada porque um comando de fluxo (disponível no Redis 5.0) é executado no Redis 3.0.



- O comando está desabilitado no DCS.

  Por motivos de segurança, alguns comandos do Redis são desabilitados no DCS. Para obter detalhes sobre comandos do Redis desativados e restritos, consulte **Compatibilidade de comandos**.

- O comando não pode ser executado na CLI da Web.

  Além dos comandos do Redis desativados e restritos, os comandos **KEYS** também são restritos na CLI da Web.

- O script LUA falha ao ser executado.

  Por exemplo, a mensagem de erro "ERR unknown command 'EVAL'" indica que a instância do DCS Redis é de uma versão inferior que não oferece suporte ao script LUA. Nesse caso, envie um tíquete de serviço para a instância a ser atualizada.

- Os comandos **CLIENT SETNAME** e **CLIENT GETNAME** falham ao serem executados.

  A instância do DCS Redis é de uma versão inferior que não oferece suporte a esses comandos. Nesse caso, envie um tíquete de serviço para a instância a ser atualizada.

- Os comandos a seguir estão desabilitados para instâncias de **cluster** do DCS Redis criadas antes de 10 de julho de 2018. Você pode atualizar essa instância enviando um tíquete de serviço.

  SINTER, SDIFF, SUNION, PFCOUNT, PFMERGE, SINTERSTORE, SUNIONSTORE, SDIFFSTORE, SMOVE, BLPOP, BRPOP, BRPOPLPUSH, ZUNIONSTORE, ZINTERSTORE, EVAL, EVALSHA, BITOP, RENAME, RENAMENX, RPOPLPUSH, MSETNX, SCRIPT LOAD, SCRIPT KILL, SCRIPT EXISTS, SCRIPT FLUSH

7.4 Por que a "permission denied" é retornada quando eu executo o comando keys na CLI da Web?

O comando **KEYS** é desabilitado na CLI da Web. Este comando só pode ser executado em redis-cli.

7.5 Como renomear comandos de alto risco?

Atualmente, você só pode renomear os comandos **COMMAND**, **KEYS**, **FLUSHDB**, **FLUSHALL**, **HGETALL**, **SCAN**, **HSCAN**, **SSCAN** e **ZSCAN**. Para obter detalhes, consulte **Renomeação de comandos**.

NOTA

- Atualmente, o Redis não suporta a desativação de comandos. Para evitar riscos ao usar os comandos anteriores, renomeie-os. Para obter detalhes sobre os comandos suportados e desabilitados no DCS, consulte **Compatibilidade de comandos**.
- 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.
- A renomeação de comando está disponível apenas para o Redis 4.0 e posterior e não para o Redis 3.0.

7.6 O DCS for Redis suporta o pipelining?

Sim.

Para instâncias do Redis Cluster, certifique-se de que todos os comandos em um pipeline sejam executados na mesma partição.

7.7 O DCS for Redis oferece suporte aos comandos INCR e EXPIRE?

Sim.

Para obter mais informações sobre a compatibilidade de comandos do Redis, consulte **Compatibilidade do comando de Redis**.

7.8 Por que um comando do Redis não entra em vigor?

Execute o comando no redis-cli para verificar se o comando tem efeito.

O seguinte descreve dois cenários:

- Cenário 1: definir e consultar o valor de uma chave para verificar se os comandos **SET** e **GET** funcionam.

  O comando **SET** é usado para definir o valor da cadeia. Se o valor não for alterado, execute os seguintes comandos no redis-cli para acessar a instância:

  ```
  192.168.2.2:6379> set key_name key_value
  OK
  192.168.2.2:6379> get key_name
  "key_value"
  192.168.2.2:6379>
  ```

- Cenário 2: se o tempo limite definido com o comando **EXPIRE** estiver incorreto, execute as seguintes operações:

  Defina o tempo limite para 10 segundos e execute o comando **TTL** para exibir o tempo restante. Como mostrado no exemplo a seguir, o tempo restante é de 7 segundos.

  ```
  192.168.2.2:6379> expire key_name 10
  (integer) 1
  192.168.2.2:6379> ttl key_name
  (integer) 7
  192.168.2.2:6379>
  ```


NOTA

Os clientes de Redis (incluindo redis-cli, clientes Jedis e clientes Python) se comunicam com o servidor Redis usando um protocolo binário.

Se os comandos do Redis forem executados corretamente no redis-cli, o problema pode estar no código de serviço. Nesse caso, crie logs no código para análise posterior.

7.9 Existe um limite de tempo para a execução de comandos do Redis? O que acontecerá se um comando atingir o tempo limite?

Os tempos limite do comando Redis podem ser controlados no lado do cliente ou do servidor.

- Os tempos limite no lado do cliente são controlados no código do cliente. Você pode determinar os tempos limite que atendem às necessidades do serviço. Por exemplo, se você usar Lettuce, um cliente Java, configure o parâmetro **timeout**.
- No lado do servidor, o parâmetro **timeout** é definido como **0** por padrão, indicando que as conexões nunca serão encerradas. Modifique a definição do parâmetro consultando **Modificação dos parâmetros de configuração**.

7.10 Posso configurar as chaves do Redis para não diferenciar maiúsculas de minúsculas?

Não. Como no Redis de código aberto, as chaves no DCS for Redis diferenciam maiúsculas de minúsculas e não podem ser configuradas para não diferenciar maiúsculas de minúsculas.

7.11 Posso exibir os comandos do Redis usados com mais frequência?

Não. O Redis não grava comandos e não suporta a visualização dos comandos usados com mais frequência.

7.12 Erros comuns da CLI da Web

1. ERR Wrong number of arguments for 'xxx' command

   Este erro indica que o comando Redis executado tem um erro de parâmetro (erro de sintaxe). Reescreva o comando consultando o protocolo de comando Redis de código aberto.

2. ERR unknown command 'xxx'

   Esse erro indica que o comando é desconhecido ou não é um comando válido definido pelo Redis. Reescreva o comando consultando o protocolo de comando Redis de código aberto.

3. ERR Unsupported command: 'xxx'

   Esse erro indica que o comando está desabilitado para instâncias do DCS Redis. Para obter detalhes, consulte **Comandos da CLI da Web**.

8 Monitoramento e alarmes

8.1 Como visualizar as conexões simultâneas atuais e o máximo de conexões de uma instância do DCS Redis?

Exibir conexões simultâneas de uma instância do DCS Redis

O número de conexões em tempo real recebidas por uma instância do DCS é uma métrica que pode ser visualizada no console. Para obter detalhes sobre como exibir as métricas, consulte **Visualização de métricas de monitoramento de DCS**.

No console do Cloud Eye, localize a métrica **Connected Clients**. Clique em  para exibir detalhes de monitoramento em um gráfico ampliado.

Especifique um intervalo de tempo para exibir a métrica em um período de monitoramento específico. Por exemplo, você pode selecionar um período de 10 minutos para exibir o número de conexões recebidas durante o período. No gráfico, você pode visualizar a tendência e o número total de conexões recebidas durante o período.

NOTA

A métrica **Connected Clients** significa o número de clientes conectados. Ela inclui conexões estabelecidas para monitoramento do sistema, sincronização de configuração e serviços, mas exclui conexões de réplicas.

Você também pode exibir as informações da sessão do cliente usando a função **gerenciamento de sessão** no console.

Exibir ou modificar o máximo de conexões de uma instância

Você pode exibir o número padrão e máximo permitido de conexões na página de criação de instância ou no **documento**.

Depois que uma instância é criada, você pode exibir ou alterar o valor de **maxclients** (o número máximo de conexões) na página **Instance Configuration** > **Parameters** da instância. (As instâncias do Proxy Cluster não têm esse parâmetro.)

Se o limite for excedido, as solicitações em excesso serão rejeitadas e as conexões expirarão.

8.2 O DCS for Redis oferece suporte a auditorias de comandos?

Não. Para garantir operações de leitura e gravação de alto desempenho, o DCS for Redis não audita comandos. Os comandos não são impressos.

8.3 O que devo fazer se os dados de monitoramento de uma instância do DCS Redis forem anormais?

Se tiver alguma dúvida sobre os dados de monitoramento de uma instância do DCS Redis, você pode acessar a instância por meio do redis-cli e executar o comando **INFO ALL** para visualizar as métricas. Para obter detalhes sobre a saída do comando **INFO ALL**, consulte **http://www.redis.io/commands/info**.

8.4 Por que a memória usada é maior que a memória disponível?

Para instâncias de DCS de nó único e principal/em espera, a memória de instância usada é medida pelo processo Redis-server. Para instâncias de DCS de cluster, a memória de cluster usada é a soma da memória usada de todas as partições no cluster.

Devido à implementação interna do redis-server de código aberto, a memória de instância usada é normalmente um pouco maior do que a memória de instância disponível.

O Redis aloca memória usando zmalloc. Ele não verifica se used_memory excede max_memory toda vez que a memória é alocada. Em vez disso, ele verifica se used_memory atual excede max_memory no início de uma tarefa periódica ou processamento de comando. Se used_memory exceder max_memory, o despejo será acionado. Portanto, as restrições da política max_memory não são implementadas em tempo real ou rigidamente. Um caso em que used_memory é maior que max_memory pode ocorrer ocasionalmente.

8.5 Por que o uso da largura de banda excede 100%?

As informações básicas sobre a métrica de uso de largura de banda são as seguintes.

ID da métrica	Nome da métrica	Descrição	Intervalo de valores	Objeto e dimensão monitorados	Período de monitoramento (dados brutos)
bandwidth_usage	Uso de largura de banda	Porcentagem da largura de banda utilizada para o limite máximo de largura de banda	0–200%	Objeto monitorado: Instâncias principal/em espera do DCS Redis 4.0 ou 5.0 Servidor Redis da instância de DCS Redis 4.0 ou 5.0 de Redis Cluster Dimensão: dcs_cluster_node	1 minuto

Uso da largura de banda = (Fluxo de entrada + Fluxo de saída)/(2 x Largura de banda máxima) x 100%

De acordo com a fórmula, o uso de largura de banda conta no fluxo de entrada e no fluxo de saída, que incluem o tráfego para replicação entre o principal e as réplicas. Portanto, o tráfego total é maior do que o tráfego de serviço normal.

Se o valor da métrica **Flow Control Times** for maior que 0, a largura de banda máxima foi alcançada e o controle de fluxo foi realizado.

No entanto, as decisões de controle de fluxo são tomadas sem considerar o tráfego para replicação entre o principal e as réplicas. Portanto, às vezes o uso de largura de banda excede 100%, mas o número de tempos de controle de fluxo é 0.

8.6 Por que a métrica de conexões rejeitadas é exibida?

Se a métrica **Rejected Connections** for exibida, verifique se o número de clientes conectados excede o número máximo permitido de conexões das instâncias.

- Para verificar o número máximo permitido de conexões, vá para a página de guia **Parameters** da instância e verifique o valor do parâmetro **maxclients**. (As instâncias do Proxy Cluster não têm esse parâmetro. Você pode exibir o número máximo de conexões na página de criação da instância.)
- Para verificar o número atual de conexões, vá para a página de guia **Performance Monitoring** da instância e verifique a métrica **Connected Clients**.

Se o número atual de conexões atingir o limite superior, você poderá ajustar o valor de **maxclients**. Se o valor de **maxclients** não puder mais ser aumentado, aumente as especificações da instância.

8.7 Por que o controle de fluxo é acionado? Como lidar com isso?

O controle de fluxo é acionado quando o tráfego usado por uma instância do Redis em um período excede a largura de banda máxima. As conexões podem ser descartadas devido ao controle de fluxo, resultando em alta latência de serviço e exceções de conexão do cliente.

NOTA

Para obter detalhes sobre a largura de banda máxima permitida, consulte a coluna "Largura de banda garantida/máxima" de diferentes tipos de instância listados em **Especificações da instância de DCS**.

Mesmo que o uso da largura de banda seja baixo, o controle de fluxo ainda pode ser acionado. O uso de largura de banda em tempo real é relatado uma vez em cada período de relatório. Os controles de fluxo são verificados a cada segundo. O tráfego pode aumentar em segundos e, em seguida, cair entre os períodos de relatório. No momento em que o uso da largura de banda é relatado, ele já pode ter sido restaurado ao nível normal.

Para instâncias principais/em espera:

- Se o controle de fluxo for sempre acionado quando o uso da largura de banda for baixo, pode haver microbursts de serviço ou teclas grandes ou de atalho. Nesse caso, verifique se há teclas grandes ou de atalho.
- Se o uso da largura de banda permanece alta, o limite de largura de banda pode ser excedido. Neste caso, expanda a capacidade. Maior capacidade suporta maior largura de banda.

Para instâncias de cluster:

- Se o controle de fluxo for acionado apenas em uma ou algumas partições, as partições podem ter teclas grandes ou de atalho.
- Se o controle de fluxo ou o alto uso da largura de banda ocorrer em todos ou na maioria das partições ao mesmo tempo, o uso da largura de banda da instância atingiu o limite. Nesse caso, expanda a capacidade da instância.

NOTA

- Execute a análise de teclas grandes e de teclas de atalho no console do DCS e tome as medidas apropriadas. Para obter detalhes, consulte **Análise de teclas grandes e teclas de atalho**.
- A execução de comandos (como **KEYS**) que consomem muitos recursos pode causar alta utilização da CPU e da largura de banda. Como resultado, o controle de fluxo é acionado.

9 Alternância entre principal/em espera

9.1 Quando ocorre uma alternância principal/em espera?

Uma alternância principal/em espera pode ocorrer nos seguintes cenários:

- Uma operação de alternância principal/em espera é iniciada no console do DCS.
- Uma alternância principal/em espera será acionada quando o nó principal de uma instância principal/em espera falhar.

  Por exemplo, se os comandos (como **KEYS**) que consomem muitos recursos forem usados ou os logs forem envelhecidos e excluídos em lotes, o uso da CPU aumentará, acionando uma alternância principal/em espera.
- Se você reiniciar uma instância principal/em espera no console do DCS, uma alternância principal/em espera será acionada.

Após ocorrer uma alternância principal/em espera, você receberá uma notificação. Verifique se os serviços do cliente estão funcionando corretamente. Caso contrário, verifique se a conexão TCP está normal e se ela pode ser restabelecida após a alternância principal/em espera para restaurar os serviços.

9.2 Como a alternância principal/em espera afeta os serviços?

Se ocorrer uma falha em uma instância de DCS principal/em espera ou de cluster, um failover será acionado automaticamente. Os serviços podem ser interrompidos por menos de meio minuto durante a detecção de exceções e o failover.

9.3 O cliente precisa alternar o endereço de conexão após uma alternância principal/em espera?

Não. Se o nó principal falhar ou se for executada uma alternância principal/em espera, o nó em espera será promovido a principal e tomará o endereço IP original.

9.4 Como funciona a replicação principal/em espera do Redis?

As instâncias principal/em espera do Redis também são chamadas de instâncias principal/secundária. Geralmente, as atualizações para o nó de cache principal são replicadas de forma automática e assíncrona para o nó de cache em espera. Isso significa que os dados no nó de cache em espera podem nem sempre ser consistentes com os dados no nó de cache principal. A inconsistência é normalmente vista quando a velocidade de gravação de I/O do nó principal é mais rápida do que a velocidade de sincronização do nó em espera ou ocorre uma latência de rede entre os nós principal e em espera. Se um failover ocorrer quando alguns dados ainda não tiverem sido replicados para o nó em espera, esses dados poderão ser perdidos após o failover.

10 Compras e permissões

10.1 Por que não consigo criar uma instância do DCS Redis ou do Memcached?

- A sub-rede não tem endereços IP suficientes.

  Análise: cada nó em uma instância de DCS deve receber um endereço IP. Portanto, uma instância de nó único requer um endereço IP, uma instância principal/em espera requer dois endereços IP e uma instância de cluster requer vários endereços IP.

  Solução: crie a instância em uma sub-rede diferente na VPC ou libere endereços IP na sub-rede atual.

- O usuário do IAM não tem as permissões necessárias para criar uma instância.

  Análise: o grupo ao qual o usuário pertence deve receber a política **DCS FullAccess** ou a função **DCS Administrator** ou outras políticas que contenham as permissões necessárias para criar instâncias do DCS.

  Solução: crie uma instância de DCS como administrador.

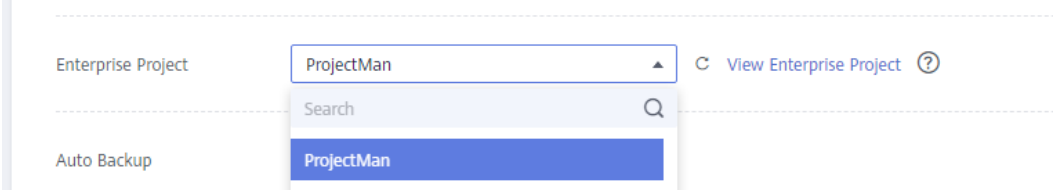
10.2 Por que não consigo exibir as informações da sub-rede e do grupo de segurança ao criar uma instância de DCS?

Isso pode ocorrer porque você não tem as funções de **Server Administrator** e **VPC Administrator**. Para obter detalhes sobre como adicionar permissões de usuário, consulte **Modificação de permissões de grupos de usuários**.

10.3 Por que não posso selecionar o projeto empresarial necessário ao criar uma instância de DCS?

Sintoma

O projeto empresarial desejado não é exibido durante a criação da instância.



Causa

O grupo de usuários não tem permissões de DCS no projeto empresarial desejado.

Solução

1. Efetue logon no console do DCS.
2. No canto superior direito, escolha **Enterprise** > **Project Management**. Na página exibida, clique em **View Resource** na linha que contém o projeto empresarial desejado.
3. Clique na guia **Permissions**. Em seguida, clique em **Authorize User Group**.

   NOTA

   Clique em **Authorize User Group** para conceder permissões a um grupo de usuários ou clique em **Authorize User** para conceder permissões a um usuário.
4. Clique em **Authorize** na linha que contém o usuário ou grupo de usuários ao qual você deseja conceder permissões.
5. Procure e selecione a política **DCS FullAccess**, clique em **Next** e clique em **OK**.

   Para obter mais informações sobre políticas de permissões de DCS, consulte **Gerenciamento de permissões**.

   NOTA

   Se você configurar o **DCS UserAccess** (contendo instruções de negação) e as políticas **DCS FullAccess**, não será possível criar, modificar, excluir ou escalar instâncias de DCS porque as instruções de negação terão precedência. Para executar as operações permitidas pelos **DCS FullAccess**, exclua primeiro **DCS UserAccess**.

10.4 Por que um usuário do IAM não pode ver uma nova instância do DCS Redis?

Sintoma

Um usuário do IAM não pode ver uma instância do DCS Redis recém-adquirida.

Possível causa

O usuário do IAM não tem permissões para o projeto empresarial ao qual a nova instância pertence.

Solução

1. Efetue logon no console do DCS.
2. No canto superior direito, escolha **Enterprise** > **Project Management**. Na página exibida, clique em **View Resource** na linha que contém o projeto empresarial desejado.

3. Clique na guia **Permissions**. Em seguida, clique em **Assign Permissions** na guia **User Groups**.
4. Selecione os grupos de usuários aos quais você deseja atribuir permissões e clique em **Next**.
5. Selecione **DCS UserAccess** e clique em **OK**.

11 Uso do Memcached

11.1 Posso despejar dados de instância do DCS Memcached para análise?

Não.

11.2 Qual versão do Memcached é compatível com o DCS for Memcached?

O DCS for Memcached é baseado no Redis 3.0 e é compatível com o Memcached 1.5.1.

11.3 Quais estruturas de dados o DCS for Memcached suporta?

Somente a estrutura de chave-valor é suportada.

11.4 O DCS for Memcached oferece suporte ao acesso público?

Não.

Se o acesso público estiver desabilitado para uma instância do DCS, você não poderá acessá-la em ambientes locais e só poderá acessá-la por meio de um ECS que esteja na mesma VPC da instância. As VPCs são usadas para garantir a segurança da rede de serviços de nuvem pública.

Durante o desenvolvimento e a depuração de aplicações, você pode se conectar a uma instância de DCS do seu ambiente local usando um ECS que pode se comunicar com sua instância para encaminhar suas solicitações. Para obter detalhes, consulte **Uso do túnel SSH para acesso público a uma instância do DCS**.

11.5 Posso modificar parâmetros de configuração de instâncias do DCS Memcached?

A configuração de parâmetros é permitida somente quando as instâncias do DCS estão no estado **Running**.

Para obter detalhes, consulte **Modificação dos parâmetros de configuração**.

11.6 Quais são as diferenças entre DCS for Memcached e Memcached auto-hospedado?

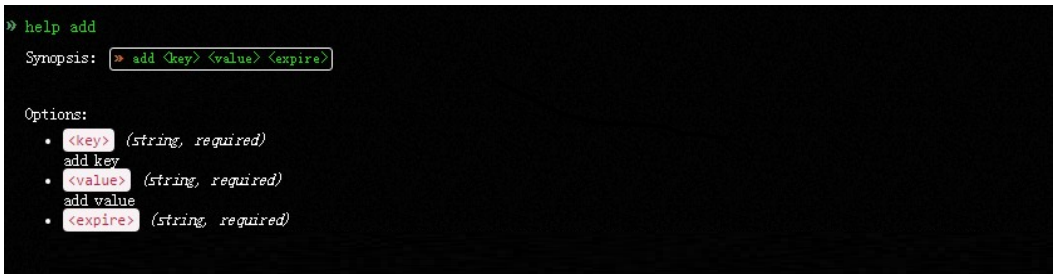
Tabela 11-1 descreve as diferenças entre o DCS for Memcached e o Memcached auto-hospedado.

Tabela 11-1 Comparar DCS for Memcached e Memcached auto-hospedado

Item	DCS for Memcached	Memcached auto-hospedado
Implementação	Fácil de implementar. O DCS for Memcached pode ser usado imediatamente sem a necessidade de se preocupar com o hardware ou software.	Envolve operações e configurações complicadas.
Disponibilidade	As instâncias principais/em espera usam espera ativa para garantir serviços estáveis. Se o nó principal estiver com defeito, o nó de cache em espera se tornará automaticamente o nó principal para evitar um único ponto de falha.	Requer configurações adicionais.
Segurança	Usa a VPC e grupos de segurança para controle de segurança de acesso à rede.	Requer que você projete e implemente um mecanismo de segurança por conta própria.
Aumento de dimensionamento	Suporta aumento de dimensionamento on-line no console.	Requer hardware adicional e reinicialização do seu serviço.

11.7 Quais políticas o DCS for Memcached usa para lidar com dados expirados?

O DCS for Memcached permite definir o tempo de expiração para dados armazenados com base nos requisitos de serviço. Por exemplo, você pode definir o tempo **expire** ao executar a operação **add**.



Por padrão, os dados não são despejados de instâncias do DCS Memcached. Na versão atual do DCS for Memcached, você pode selecionar uma política de despejo.

Para obter detalhes sobre os seis tipos de políticas de remoção de dados, consulte **Qual é a política padrão de despejo de dados?**

11.8 Como selecionar as AZs ao criar uma instância do DCS Memcached?

Diferentes AZs dentro de uma região não diferem em funções.

Geralmente, a implementação de instâncias em uma AZ apresenta menor latência de rede, enquanto a implementação entre AZs garante a recuperação de desastres. Se sua aplicação exigir menor latência de rede, escolha a implementação de AZ única.

DCS for Memcached suporta a implementação entre AZs. Ao criar uma instância do DCS Memcached no console do DCS, você pode selecionar qualquer AZ na mesma região do ECS para comunicação entre o ECS e a instância. Para uma menor latência de rede, selecione a mesma AZ do ECS.

Suponha que você tenha um ECS que esteja em **AZ B** na região **CN-Hong Kong**. Ao comprar uma instância do DCS Memcached, você pode selecionar qualquer AZ na **CN-Hong Kong**. Se você selecionar **AZ B** em **CN-Hong Kong**, sua instância poderá se comunicar com o ECS com menor latência de rede.

Observe que pode haver apenas uma AZ disponível devido a recursos insuficientes quando você cria uma instância do DCS Memcached. Isso não afeta o uso normal de DCS.

