Distributed Cache Service

Preguntas frecuentes

Edición 01

Fecha 2025-01-20

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Marcas registradas y permisos

El logotipo HUAWEI y otras marcas registradas de Huawei pertenecen a Huawei Technologies Co., Ltd.
Todas las demás marcas registradas y los otros nombres comerciales mencionados en este documento son propiedad de sus respectivos titulares.

Aviso

Es posible que la totalidad o parte de los productos, las funcionalidades y/o los servicios que figuran en el presente documento no se encuentren dentro del alcance de un contrato vigente entre Huawei Cloud y el cliente. Las funcionalidades, los productos y los servicios adquiridos se limitan a los estipulados en el respectivo contrato. A menos que un contrato especifique lo contrario, ninguna de las afirmaciones, informaciones ni recomendaciones contenidas en el presente documento constituye garantía alguna, ni expresa ni implícita.

Huawei está permanentemente preocupada por la calidad de los contenidos de este documento; sin embargo, ninguna declaración, información ni recomendación aquí contenida constituye garantía alguna, ni expresa ni implícita. La información contenida en este documento se encuentra sujeta a cambios sin previo aviso.

Huawei Cloud Computing Technologies Co., Ltd.

Dirección: Huawei Cloud Data Center Jiaoxinggong Road
Avenida Qianzhong
Nuevo distrito de Gui'an
Gui Zhou, 550029
República Popular China

Sitio web: https://www.huaweicloud.com/intl/es-us/

Índice

1 Tipos/Versiones de Instancias

1.1 Comparación entre Redis y Memcached

Redis y Memcached son las bases de datos en memoria de código abierto populares que son fáciles de usar y proporcionan un mayor rendimiento que las bases de datos relacionales.

¿Cómo puedo seleccionar entre las dos bases de datos de clave-valor?

Memcached es adecuado para almacenar las estructuras de datos simples, mientras que Redis es adecuado para almacenar datos más complejos y más grandes que requieren persistencia.

Para más detalles, mira la siguiente tabla.

Tabla 1-1 Diferencias entre Redis y Memcached

Concepto	Redis	Memcached
Latencia	Base de datos en memoria con la latencia de submilisegundos	Base de datos en memoria con la latencia de submilisegundos
Facilidad de uso	Sintaxis sencilla y fácil de usar	Sintaxis sencilla y fácil de usar
Almacenamiento distribuido	Expansión horizontal en modo de clúster	Admitido
Cliente multilingüe	Admite conexiones de clientes en más de 30 idiomas, incluidos Java, C y Python.	Admite conexiones de clientes en más de 10 idiomas, incluidos Java, C y Python.

Concepto	Redis	Memcached
Hilo/Proceso	Un solo núcleo y un solo hilo Comunicación de subproceso único, evitando la conmutación y la contención innecesarias del contexto La E/S sin bloqueo (multiplexación de E/S) se utiliza para reducir el consumo de recursos cuando se conectan varios clientes.	Varios hilos y escalable El rendimiento de Memcached se puede mejorar aumentando el número de CPU. Hay una ventaja de rendimiento obvia en el escenario donde el valor de la clave es grande.
Almacenamiento persistente	Admitido Cada operación de escritura (agregar, eliminar o modificar datos) se puede grabar en el disco (archivo AOF).	Admitido **NOTA** La persistencia no es compatible con Memcached de código abierto, pero es compatible con Huawei Cloud DCS for Memcached.
Estructura de datos	Soporta estructuras de datos complejas como hash, lista, conjunto y conjunto ordenado, atendiendo a varios escenarios.	Soporta las cadenas simples.
Soporte de scripts de Lua	Admitido	No admitido
Copia de respaldo de la instantánea	Admitido Las instantáneas se generan periódicamente. Por lo tanto, no hay garantía de que los datos no se pierdan. Redis bifurca un subproceso para generar instantáneas. Cuando hay una gran cantidad de datos, el servicio Redis puede interrumpirse en un corto tiempo.	No admitido
Migración de datos	Admitido Los datos se pueden realizar copias de respaldo y migrar a una nueva instancia de Redis mediante la restauración de instantáneas de RDB o la reproducción de los archivos de AOF.	No admitido
Restricción del valor clave	El valor de una clave puede ser de hasta 1 GB.	1 MB

Concepto	Redis	Memcached
Bases de datos múltiples	Una instancia de DCS Redis de nodo único o principal/en espera admite hasta 256 bases de datos de Redis. Una instancia de Clúster Proxy o de Clúster Redis soporta solo una base de datos, es decir, DB0.	No admitido

En base a la comparación anterior, tanto Redis como Memcached son fáciles de usar y tienen un alto rendimiento. Sin embargo, Redis y Memcached son diferentes en cuanto al almacenamiento de estructura de datos, persistencia, respaldo, migración y compatibilidad con scripts. Se recomienda seleccionar el motor de caché más adecuado en función de los escenarios reales de la aplicación.

NOTA

Memcached es adecuado para escenarios de almacenamiento en caché de pequeña cantidad de datos estáticos, donde los datos solo se leen sin más cómputo y procesamiento, por ejemplo, fragmentos de código HTML.

Redis tiene las estructuras de datos más ricas y los escenarios de aplicación más amplios.

1.2 Comparación de versiones

Al crear una instancia de DCS compatible con Redis, puede seleccionar la versión del motor de caché y el tipo de instancia.

NOTA

DCS for Redis 3.0 ya no se proporciona. Puede utilizar DCS for Redis 4.0, 5.0 o 6.0 en su lugar.

- **Versión**

  DCS soporta Redis 6.0, 5.0, 4.0 y 3.0. **Tabla 1-2** describe las diferencias entre estas versiones. Para obtener más información sobre las nuevas características de Redis 4.0 y 5.0, consulte las **Nuevas características de DCS for Redis 4.0** y **Nuevas características de DCS for Redis 5.0** .

Tabla 1-2 Diferencias entre las versiones de Redis

Característica	Redis 3.0	Redis 4.0 & Redis 5.0	Redis 6.0
Compatibilidad con software de código abierto:	Redis 3.0.7	Redis 4.0.14 y 5.0.14, respectivamente **NOTA** Las instancias de DCS Redis 5.0 creadas antes de febrero de 2022 son compatibles con Redis 5.0.9 de código abierto. ● Para obtener detalles sobre cómo consultar la versión de código abierto, consulte **¿Cómo puedo ver la versión de una instancia de DCS Redis?** ● Para usar Redis 5.0.14, cree otra instancia. Actualmente, la versión de Redis no se puede actualizar.	Edición básica: Redis 6.2.7 Edición profesional: KeyDB 6.0.16
Modo de despliegue de instancia	Basado en las máquinas virtuales	Containerizado basado en los servidores físicos	Containerizado basado en los servidores físicos
Arquitectura de CPU	x86	x86	x86
Tiempo necesario para crear una instancia	3–15 minutos o 10–30 minutos para las instancias de clúster.	8 segundos	8 segundos
QPS	100,000 QPS por nodo	100,000 QPS por nodo	Edición básica: 150,000 QPS por nodo Edición profesional: 400,000 QPS por nodo
Acceso a la red pública	Admitido	No admitido	No admitido
Conexión de nombre de dominio	Compatible con VPC	Compatible con VPC	Compatible con VPC

Característica	Redis 3.0	Redis 4.0 & Redis 5.0	Redis 6.0
Gestión de datos visualizados	No admitido	Acceso a Web CLI para Redis y gestión de datos.	Acceso a Web CLI para Redis y gestión de datos.
Tipo de instancia	Nodo único, principal/en espera y Clúster Proxy	Nodo único, principal/en espera, Clúster Proxy y Clúster Redis	Solo principal/en espera
Memoria total de instancia	Rango de 2 GB a 1024 GB.	Las especificaciones regulares van desde 2 GB hasta 1024 GB. Las pequeñas especificaciones de 128 MB, 256 MB, 512 MB y 1 GB también están disponibles para las instancias de nodo único y principal/en standby.	4 GB, 8 GB, 16 GB, 32 GB y 64 GB (128 MB, 256 MB, 512 MB y 1 GB se admiten adicionalmente para instancias de nodo único y principal/en espera)
Ampliación/ reducción de la capacidad	Ampliación y reducción de la capacidad en línea	Ampliación y reducción de la capacidad en línea	Ampliación y reducción de la capacidad en línea
Copia de respaldo y restauración	Compatible con las instancias principal/en standby y de Clúster Proxy	Compatible con las instancias principal/standby, de Clúster Proxy y de Clúster Redis	Compatible con las instancias principal/en standby

NOTA

Las arquitecturas subyacentes varían según la versión de Redis. Una vez que se elige una versión de Redis, no se puede cambiar. Por ejemplo, no puede actualizar una instancia de DCS Redis 3.0 a Redis 4.0 o 5.0. Si necesita una versión de Redis superior, compre una nueva instancia que cumpla con sus requisitos y, a continuación, migre los datos de la instancia antigua a la nueva.

- **Tipo de instancia**

  DCS proporciona los tipos de instancia de nodo único, principal/en espera, de Clúster Proxy y de Clúster Redis. Para obtener más información sobre sus arquitecturas y escenarios de aplicación, consulte **Tipos de instancia de DCS**.

1.3 Nuevas características de DCS for Redis 4.0

En comparación con DCS for Redis 3.0, DCS for Redis 4.0 y las versiones posteriores agregan soporte para las nuevas características de Redis de código abierto y admite la creación de instancias más rápida.

El despliegue de instancia cambió del modo VM al modo de contenedorización basado en servidor físico. Una instancia se puede crear en un plazo de 8 a 10 segundos.

Redis 4.0 ofrece las siguientes características nuevas:

1. Nuevos comandos, como **MEMORY** y **SWAPDB**
2. Lazyfree, retrasando la eliminación de claves grandes y reduciendo el impacto de la eliminación en los recursos del sistema
3. Optimización del rendimiento de la memoria, es decir, desfragmentación activa

Comando MEMORY

En Redis 3.0 y las versiones anteriores, puede ejecutar el comando **INFO MEMORY** para aprender solo las estadísticas de memoria limitadas. Redis 4.0 introduce el comando **MEMORY** para ayudarle a comprender mejor el uso de la memoria de 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]>
```

uso

Ingrese **memory usage *[key]***. Si la clave existe, se devuelve la memoria estimada utilizada por el valor de la clave. Si la clave no existe, se devuelve **nil**.

```
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. **usage** collects statistics on the memory usage of the value and the key, excluding the Expire memory usage of the key.

```
// 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. For hashes, lists, sets, and sorted sets, the **MEMORY USAGE** command samples statistics and provides the estimated memory usage.

   Usage: **memory usage** *keyset* **samples** *1000*

   keyset indicates the key of a set, and *1000* indicates the number of samples.

stats

Returns the detailed memory usage of the current instance.

Usage: **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"
 ... ...
```

The following table describes the meanings of some return items.

Tabla 1-3 MEMORY STATS return values

Return Value	Description
peak.allocated	Peak memory allocated by the allocator during Redis instance running. It is the same as **used_memory_peak** of **info memory**.
total.allocated	The number of bytes allocated by the allocator. It is the same as **used_memory** of **info memory**
startup.allocated	Initial amount of memory consumed by Redis at startup in bytes
replication.backlog	Size in bytes of the replication backlog. It is specified in the **repl-backlog-size** parameter. The default value is **1 MB**.
clients.slaves	The total size in bytes of all replicas overheads

Return Value	Description
clients.normal	The total size in bytes of all clients overheads
overhead.total	The sum of all overheads. **overhead.total** is the total memory **total.allocated** allocated by the allocator minus the actual memory used for storing data.
keys.count	The total number of keys stored across all databases in the server
keys.bytes-per-key	Average number of bytes occupied by each key. Note that the overhead is also allocated to each key. Therefore, this value does not indicate the average key length.
dataset.bytes	Memory bytes occupied by Redis data, that is, **overhead.total** subtracted from **total.allocated**
dataset.percentage	The percentage of **dataset.bytes** out of the net memory usage
peak.percentage	The percentage of peak.allocated out of **total.allocated**
fragmentation	Memory fragmentation rate

doctor

Usage: **memory doctor**

If the value of **used_memory** (**total.allocated**) is less than 5 MB, **MEMORY DOCTOR** considers that the memory usage is too small and does not perform further diagnosis. If any of the following conditions is met, Redis provides diagnosis results and suggestions:

1. The peak allocated memory is greater than 1.5 times of the current **total_allocated**, that is, **peak.allocated**/**total.allocated** > 1.5, indicating that the memory fragmentation rate is high, and that the RSS is much larger than **used_memory**.
2. The value of high fragmentation/fragmentation is greater than 1.4, indicating that the memory fragmentation rate is high.
3. The average memory usage of each normal client is greater than 200 KB, indicating that the pipeline may be improperly used or the Pub/Sub client does not process messages in time.
4. The average memory usage of each slave client is greater than 10 MB, indicating that the write traffic of the master is too high.

purge

Usage: **memory purge**

Executes the **jemalloc** internal command to release the memory. The released objects include the memory that is occupied but not used by Redis processes, that is, memory fragments.

NOTA

MEMORY PURGE applies only to the Redis instance that uses **jemalloc** as the allocator.

Lazyfree

Problem

Redis is single-thread. When a time-consuming request is executed, all requests are queued. Before the request is completed, Redis cannot respond to other requests. As a result, performance problems may occur. One of the time-consuming requests is deleting a large key.

Principle

The Lazyfree feature of Redis 4.0 avoids the blockage caused by deleting large keys, ensuring performance and availability.

When deleting a key, Redis asynchronously releases the memory occupied by the key. The key release operation is processed in the sub-thread of the background I/O (BIO).

Usage

1. Active deletion
   - **UNLINK**

     Similar to **DEL**, this command removes keys. If there are more than 64 elements to be deleted, the memory release operation is executed in an independent BIO thread. Therefore, the **UNLINK** command can delete a large key containing millions of elements in a short time.
   - **FLUSHALL** and **FLUSHDB**

     An **ASYNC** option was added to **FLUSHALL** and **FLUSHDB** in order to let the entire dataset or a single database to be freed asynchronously.
2. Passive deletion: deletion of expired keys and eviction of large keys

   There are four scenarios for passive deletion and each scenario corresponds to a parameter. These parameters are disabled by default.

```
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.
```

Other New Commands

1. **swapdb**

   Swaps two Redis databases.

   swapdb *dbindex1 dbindex2*
2. **zlexcount**

   Returns the number of elements in the sorted set.

   zlexcount *key min max*

Memory and Performance Optimization

1. Compared to before, the same amount of data can be stored with less memory.
2. Used memory can be defragmented and gradually evicted.

1.4 Nuevas características de DCS for Redis 5.0

DCS for Redis 5.0 es compatible con las nuevas características de Redis 5.0 de código abierto, además de todas las mejoras y nuevos comandos de Redis 4.0.

Estructura de datos de flujo

Flujo es un nuevo tipo de datos introducido con Redis 5.0. Soporta persistencia de mensajes y multidifusión.

Figura 1-1 muestra la estructura de un flujo de Redis, que permite que se agregan mensajes al flujo.

Flujos tienen las siguientes características:

1. Un flujo puede tener varios grupos de consumidores.
2. Cada grupo de consumidores contiene un **Last_delivered_id** que apunta al último artículo consumido (mensaje) en el grupo de consumidores.
3. Cada grupo de consumidores contiene múltiples consumidores. Todos los consumidores comparten el **last_delivered_id** del grupo de consumidores. Un mensaje puede ser consumido por un solo consumidor.
4. **pending_ids** en el consumidor puede ser utilizado para registrar los ID de los artículos que han sido enviados al cliente, pero no han sido reconocidos.
5. Para una comparación detallada entre flujo y otras estructuras de datos de Redis, consulte **Tabla 1-4**.

Figura 1-1 Estructura de datos de flujo

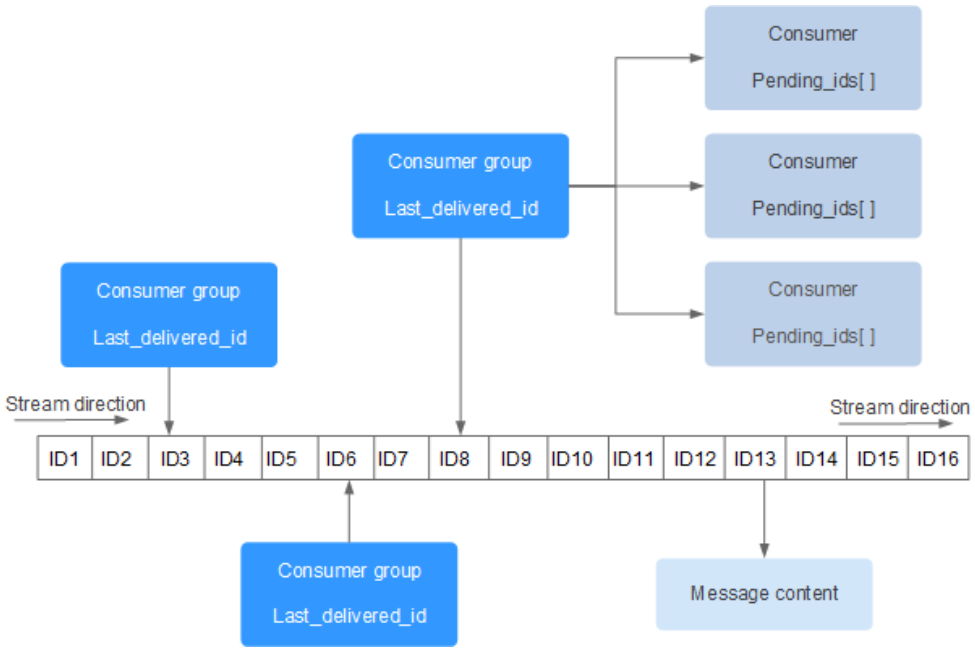


Tabla 1-4 Diferencias entre flujo y las estructuras de datos de Redis existentes

Concepto	Flujo	List, Pub/Sub, Zset
Complejidad de buscar artículos	O(log(N))	List: O(N)
Desfase	Admitido. Cada elemento tiene un ID único. El ID no se cambia a medida que se agregan o desalojan otros elementos.	List: no soportado. Si se desaloja un artículo, no se puede localizar el último artículo.
Persistencia	Admitido. Flujos se mantienen en archivos de AOF y de RDB.	Pub/Sub: No soportado.
Grupo de consumidores	Admitido.	Pub/Sub: No soportado.
Confirmación	Admitido.	Pub/Sub: No soportado.
Rendimiento	No relacionado con el número de consumidores.	Pub/Sub: Positivamente relacionado con el número de clientes.
Desalojo	Flujos son eficientes en la memoria al bloquear para desalojar los datos que son demasiado antiguos y usar un árbol de radix y un paquete de lista.	Zset consume más memoria porque no admite la inserción de los mismos elementos, el bloqueo o la expulsión de datos
Eliminación de elementos aleatoria	No se admite.	Zset: Admitido.

Comandos de flujo

Los comandos de flujo se describen a continuación en el orden en que se usan. Para obtener más información, véase **Tabla 1-5**.

1. Ejecute el comando **XADD** para agregar un elemento de flujo, es decir, crear un flujo. El número máximo de mensajes que se pueden guardar se puede especificar al agregar el elemento.
2. Cree un grupo de consumidores ejecutando el comando **XGROUP**.
3. Un consumidor utiliza el comando **XREADGROUP** para consumir mensajes.
4. Después del consumo, el cliente ejecuta el comando **XACK** para confirmar que el consumo es exitoso.

Figura 1-2 Comandos de flujo

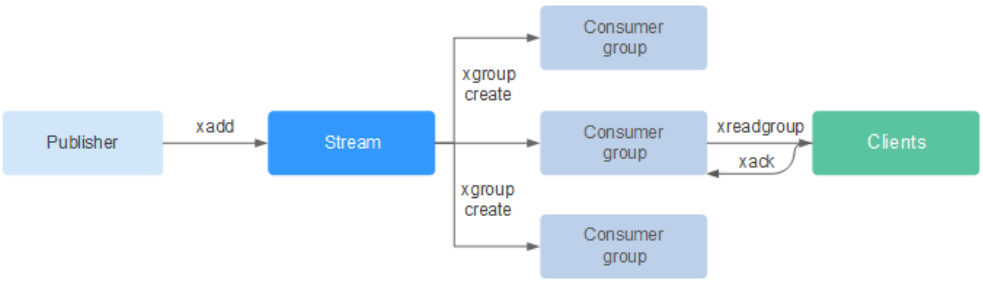


Tabla 1-5 Descripción de los comandos de flujo

Comando	Descripción	Sintaxis
XACK	Elimina uno o varios mensajes del *pending entry list* (PEL) de un grupo de consumidores del flujo.	XACK key group ID [ID ...]
XADD	Agrega una entrada especificada a flujo en una clave especificada. Si la clave no existe, ejecutar este comando dará como resultado una clave que se creará automáticamente en función de la entrada.	XADD key ID field string [field string ...]
XCLAIM	Cambia la propiedad de un mensaje pendiente, de modo que el nuevo propietario sea el consumidor especificado como argumento de comando.	XCLAIM key group consumer min-idle-time ID [ID ...] [IDLE ms] [TIME ms-unix-time] [RETRYCOUNT count] [FORCE] [JUSTID]
XDEL	Elimina las entradas especificadas de un flujo y devuelve el número de entradas eliminadas, que puede ser diferente del número de ID pasados al comando en caso de que no existan ciertos ID.	XDEL key ID [ID ...]
XGROUP	Gestiona los grupos de consumidores asociados a un flujo. Puede usar **XGROUP** para: ● Crear un nuevo grupo de consumidores asociado a una secuencia. ● Destruir un grupo de consumidores. ● Eliminar un consumidor especificado de un grupo de consumidores. ● Establecer el *last delivery ID* del grupo de consumidores a otra cosa.	XGROUP [CREATE key groupname id-or-$] [SETID key id-or-$] [DESTROY key groupname] [DELCONSUMER key groupname consumername]

Comando	Descripción	Sintaxis
XINFO	Recupera información diferente sobre los flujos y los grupos de consumidores asociados.	XINFO [CONSUMERS key groupname] key key [HELP]
XLEN	Devuelve el número de entradas de un flujo. Si la clave especificada no existe, se devuelve **0**, lo que indica una secuencia vacía.	XLEN key
XPENDING	Obtiene datos de un flujo a través de un grupo de consumidores. Este comando es la interfaz para inspeccionar la lista de mensajes pendientes con el fin de observar y comprender qué clientes están activos, qué mensajes están pendientes de ser consumidos, o para ver si hay los mensajes inactivos.	XPENDING key group [start end count] [consumer]
XRANGE	Devuelve entradas que coinciden con un rango dado de ID.	XRANGE key start end [COUNT count]
XREAD	Lee datos de uno o varios flujos, solo devuelve entradas con un ID mayor que el último ID recibido informado por la persona que llama.	XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]
XREADGROUP	Una versión especial del comando **XREAD**, que se utiliza para especificar un grupo de consumidores para leer.	XREADGROUP GROUP group consumer [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]
XREVRANGE	Este comando es exactamente igual a **XRANGE** pero con la notable diferencia de devolver las entradas en el orden inverso, y también tomar el rango de inicio-final en el orden inverso.	XREVRANGE key end start [COUNT count]
XTRIM	Recorta la secuencia a un número especificado de elementos, si es necesario, desalojando los elementos antiguos (elementos con identificadores inferiores).	XTRIM key MAXLEN [~] count

Confirmación de mensaje (elemento de flujo)

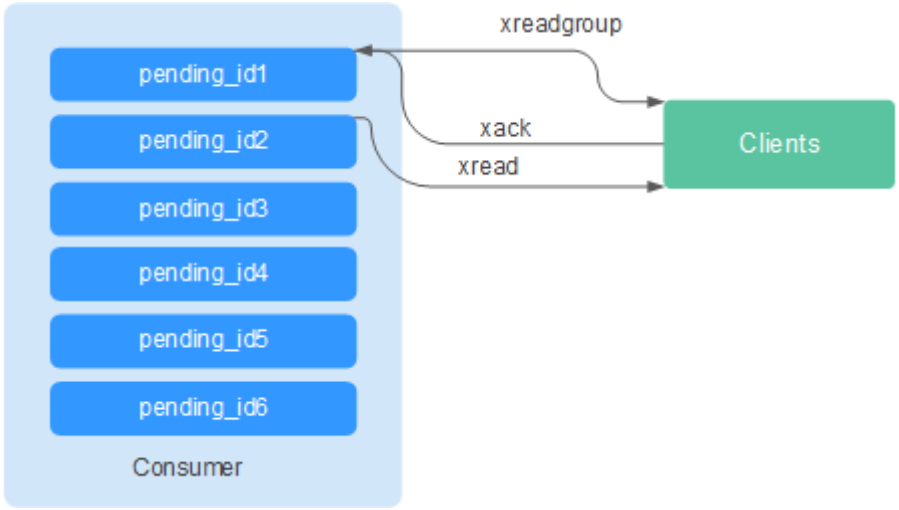
En comparación con Pub/Sub, las transmisiones no solo admiten grupos de consumidores, sino también la confirmación de mensajes.

Cuando un consumidor invoca el commando **XREADGROUP** para leer o invoca el commando **XCLAIM** para asumir un mensaje, el servidor no sabe si el mensaje se procesa al

menos una vez. Por lo tanto, una vez que se ha procesado con éxito un mensaje, el consumidor debe invocar el commando **XACK** para notificar el flujo de manera que el mensaje no se procesará de nuevo. Además, el mensaje se elimina de PEL y la memoria se liberará del servidor de Redis.

En algunos casos, como las fallas de red, el cliente no invoca **XACK** después del consumo. En tales casos, el ID del elemento se conserva en PEL. Después de volver a conectar el cliente, establezca el ID de mensaje de inicio de **XREADGROUP** en 0-0, indicando que todos los mensajes PEL y mensajes después de **last_id** son leídos. Además, se debe admitir la transmisión de mensajes repetidos cuando los consumidores consumen mensajes.

Figura 1-3 Mecanismo de confirmación



Optimización del uso de memoria

El uso de memoria de Redis 5.0 está optimizado en función de la versión anterior.

- Desfragmentación activa

  Si una clave se modifica con frecuencia y la longitud del valor cambia constantemente, Redis asignará memoria adicional para la clave. Para lograr un alto rendimiento, Redis utiliza el asignador de memoria para gestionar la memoria. La memoria no siempre se libera hasta el sistema operativo. Como resultado, se producen los fragmentos de memoria. Si la relación de fragmentación (**used_memory_rss**/**used_memory**) es superior a 1.5, el uso de memoria es ineficiente.

  Para reducir los fragmentos de memoria, planifique y use adecuadamente los datos de caché y estandarice la escritura de datos.

  Para Redis 3.0 y las versiones anteriores, los problemas de fragmentación de memoria se resuelven reiniciando el proceso regularmente. Se recomienda que los datos de caché reales no excedan el 50% de la memoria disponible.

  Para Redis 4.0, se admite la desfragmentación activa y la memoria se desfragmenta mientras está en línea. Además, Redis 4.0 admite la desfragmentación manual de memoria ejecutando el comando **memory purge**.

  Para Redis 5.0, se admite una desfragmentación activa mejorada con Jemalloc actualizado, que es más rápido, más inteligente y proporciona una latencia más baja.

- Mejoras en el despliegue de HyperLogLog

HyperLogLog es una estructura de datos probabilística utilizada para calcular la cardinalidad de un conjunto mientras consume poca memoria. Redis 5.0 mejora HyperLogLog al optimizar aún más su uso de memoria.

Por ejemplo: el árbol B es eficiente en el recuento, pero consume mucha memoria. Utilizando HyperLogLog se puede guardar mucha memoria. Mientras que el árbol B requiere 1 MB de memoria para contar, HyperLogLog solo necesita 1 KB.

- Estadísticas de memoria mejoradas

  La información devuelta por el comando **INFO** es más detallada.

Nuevos y mejores comandos

1. **Gestión de clientes mejorada**
   - redis-cli admite la gestión de clústeres.

     En Redis 4.0 y versiones anteriores, es necesario instalar el módulo **redis-trib** para gestionar clústeres.

     Redis 5.0 optimiza redis-cli, integrando todas las funciones de gestión de clústeres. Puede ejecutar el comando **redis-cli --cluster help** para obtener más información.
   - El rendimiento del cliente se mejora en escenarios frecuentes de conexión y desconexión.

     Esta optimización es valiosa cuando su aplicación necesita usar conexiones cortas.
2. **Uso más sencillo de conjuntos ordenados**

   Los comandos **ZPOPMIN** y **ZPOPMAX** se agregan para conjuntos ordenados.
   - ZPOPMIN key [count]

     Elimina y devuelve hasta miembros **count** con las puntuaciones más bajas en el conjunto ordenado almacenado en **key**. Al devolver varios elementos, el que tenga la puntuación más baja será el primero, seguido por los elementos con puntuaciones más altas.
   - ZPOPMAX key [count]

     Elimina y devuelve hasta miembros de **count** con las puntuaciones más altas en el conjunto ordenado almacenado en **key**. Al devolver varios elementos, el que tenga la puntuación más baja será el primero, seguido por los elementos con puntuaciones más bajas.
3. **Agregados más subcomandos en el comando help**

   El comando **help** se puede utilizar para ver información de ayuda, ahorrándole el problema de visitar **redis.io** cada vez. Por ejemplo, ejecute el siguiente comando para ver la información de ayuda de la secuencia **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. **Consejos de entrada de comandos de redis-cli**

   Después de escribir un comando completo, redis-cli muestra una sugerencia de parámetro que le ayudará a memorizar el formato de sintaxis del comando.

   Como se muestra en la siguiente figura, ejecute el comando **zadd** y redis-cli muestra la sintaxis de **zadd** en color claro.


```
# Cluster
cluster_enabled:0

# Keyspace
db0:keys=1,expires=0,avg_ttl=0
198.19.59.199:6379> zadd key [NX|XX] [CH] [INCR] score member [score member ...]
```

Almacenamiento en RDB de LFU y de LRU

En Redis 5.0, se agregaron las políticas de desalojo de claves de almacenamiento **LRU** y **LFU** al archivo de instantáneas de RDB.

- FIFO: El primero en entrar, el primero en salir. Los primeros datos almacenados se desalojan primero.
- LRU: Menos utilizado recientemente. Los datos que no se utilizan durante mucho tiempo se desalojan primero.
- LFU: Menos frecuentemente utilizado. Los datos que se utilizan con menos frecuencia se desalojan primero.

NOTA

El formato de archivo de RDB de Redis 5.0 se modifica y es compatible con las versiones anteriores. Por lo tanto, si se utiliza una instantánea para la migración, los datos se pueden migrar desde las versiones anteriores de Redis a Redis 5.0, pero no se pueden migrar desde Redis 5.0 a las versiones anteriores.

1.5 ¿Cuáles son las diferencias entre DCS basado en Arm y basado en x86 para Redis?

DCS for Redis es totalmente compatible con las arquitecturas de CPU Arm y x86. No difieren en funciones ni en compatibilidad con el cliente.

Sin embargo, Kunpeng y x86 difieren en los siguientes aspectos:

- Versiones de Redis compatibles
  - Redis basado en Arm: Redis 4.0 y Redis 5.0
  - Redis basado en x86: Redis 6.0, 5.0, 4.0 y 3.0
- Tipos de instancia admitidos
  - Arm: Nodo único, principal/en espera y Clúster Redis
  - x86: Nodo único, principal/en espera, Clúster Proxy de Redis 3.0 y Clúster Redis de Redis 4.0 o 5.0
- Precios

  Redis basado en Kunpeng es un 30% más barato que Redis basado en x86.
- Rendimiento

  El rendimiento de las especificaciones diferentes de instancia se enumera en **Especificaciones de instancia de DCS**.

Redis basado en x86 proporciona un mayor rendimiento de CPU única que Redis basado en Arm en los escenarios que involucran comandos complejos, como claves grandes o claves cuya complejidad temporal es mayor que O(N).

En conclusión, tanto Redis basado en Arm como Redis basado en x86 proporcionan un rendimiento que es capaz de satisfacer sus requisitos de servicio, pero Redis basado en Arm es más rentable.

1.6 ¿Puedo cambiar la arquitectura de la CPU?

No.

Si desea utilizar una arquitectura de CPU diferente, cree otra instancia con la arquitectura de CPU deseada y, a continuación, migre los datos.

NOTA

- La conmutación IP solo es compatible con las instancias de DCS Redis 4.0 y 5.0.
- La conmutación IP solo se admite cuando las instancias de origen y destino son instancias de Redis en la nube.

Requisitos previos

- La instancia de destino está disponible. Si ya tiene una instancia de DCS Redis, utilícela directamente y borre los datos de instancia antes de la migración. Para obtener más información, consulte la sección **Borrar datos de instancia de DCS**.

  Si los datos de la instancia de destino no se borran antes de la migración y las instancias de origen y destino contienen la misma clave, la clave de la instancia de destino se sobrescribirá después de la migración.
- El Redis de destino, el Redis de origen y los recursos de tareas de migración están en la misma VPC.

  NOTA

  Si las instancias de Redis de destino y de origen no están en la misma VPC, asegúrese de que los recursos de la máquina virtual de la tarea de migración puedan acceder a estas instancias.

  - Si las instancias de Redis de origen y destino están en la misma región, cree una interconexión de VPC haciendo referencia a **Interconexión de VPC**.
  - Si las instancias de origen y destino de Redis se encuentran en diferentes regiones, cree una conexión a la nube consultando **Pasos iniciales de Cloud Connect**.
- Las instancias de destino y origen utilizan el mismo puerto.
- La conmutación IP solo se puede realizar cuando se cumplen las siguientes condiciones:
  - La conmutación IP depende de la función de migración de datos. Por lo tanto, las instancias de origen y destino deben admitir la función de migración de datos.
  - En la siguiente tabla se enumeran los escenarios de conmutación IP admitidos.

Tabla 1-6 Escenarios de conmutación de IP

Fuente	Objetivo
Nodo único, separación de lectura/escritura, o principal/en espera	Nodo único, separación de lectura/escritura, principal/en espera o Clúster Proxy

Fuente	Objetivo
Clúster Proxy	Nodo único, separación de lectura/escritura, principal/en espera o Clúster Proxy

Precauciones para la conmutación de IP

1. La migración en línea se detendrá durante la conmutación.
2. Las instancias serán de solo lectura durante un minuto y se desconectarán durante varios segundos durante la conmutación.
3. Si la aplicación no puede volver a conectarse a Redis o manejar excepciones, es posible que tenga que reiniciar la aplicación después de la conmutación de IP.
4. Si las instancias de origen y destino se encuentran en diferentes subredes, la información de la subred se actualizará después de la conmutación.
5. Si la fuente es una instancia principal/en standby, la dirección IP del nodo en standby no se conmutará. Asegúrese de que sus aplicaciones no utilicen esta dirección IP.
6. Si sus aplicaciones usan un nombre de dominio para conectarse a Redis, el nombre de dominio se usará para la instancia de origen. Seleccione **Yes** para **Switch Domain Name**.
7. Asegúrese de que las contraseñas de las instancias de origen y de destino sean las mismas. Si son diferentes, la verificación fallará después de la conmutación.
8. Si se configura una lista blanca para la instancia de origen, asegúrese de que la misma lista blanca está configurada para la instancia de destino antes de cambiar las direcciones IP.

Conmutación de direcciones IP

Paso 1 Inicie sesión en la **consola DCS**..

Paso 2 Haga clic en  en la esquina superior izquierda de la consola de gestión y seleccione la región donde se encuentra la instancia.

Paso 3 En el panel de navegación, elija **Data Migration**.

Paso 4 Haga clic en **Create Online Migration Task**.

Paso 5 Introduzca el nombre y la descripción de la tarea.

Paso 6 Configure la VPC, la subred y el grupo de seguridad para la tarea de migración.

La VPC, la subred y el grupo de seguridad facilitan la migración. Asegúrese de que los recursos de migración puedan acceder a las instancias de Redis de origen y destino.

Paso 7 Configure la tarea de migración haciendo referencia a **Configuración de la tarea de migración en línea**. Establezca **Migration Type** en **Full + Incremental**.

Paso 8 En la página **Online Migration**, cuando el estado de la tarea de migración cambie a **Incremental migration in progress**, elija **More** > **Switch IP** en la columna **Operation**.

Paso 9 En el cuadro de diálogo **Switch IP**, seleccione si desea cambiar el nombre de dominio.

NOTA

- Si se utiliza un nombre de dominio, cámbielo o debe modificar el nombre de dominio en el cliente.
- Si no se utiliza ningún nombre de dominio, se actualizará el DNS de las instancias.

Paso 10 Haz clic en **OK**. La tarea de conmutación de direcciones IP se envía correctamente. Cuando el estado de la tarea de migración cambia a **IP switched**, se completa el cambio de dirección IP.

----Fin

Retroceder las direcciones IP

Si desea cambiar la dirección IP de la instancia a la dirección IP original, realice las siguientes operaciones:

Paso 1 Inicie sesión en la **consola DCS**..

Paso 2 Haga clic en en la esquina superior izquierda de la consola de gestión y seleccione la región donde se encuentra la instancia.

Paso 3 En el panel de navegación, elija **Data Migration**.

Paso 4 En la página **Online Migration**, busque la fila que contiene la tarea de migración en el estado **IP switched**, elija **More** > **Roll Back IP**.

Paso 5 En la ventana de confirmación, haga clic en **Yes**. La tarea de reversión de direcciones IP se envía correctamente. Cuando el estado de la tarea cambia a **IP rolled back**, se completa la reversión.

----Fin

1.7 ¿Cuáles son las especificaciones de la CPU de las instancias de DCS?

Al utilizar DCS, solo necesita prestar atención a los indicadores críticos como QPS, ancho de banda y memoria. No necesita preocuparse por las especificaciones de la CPU.

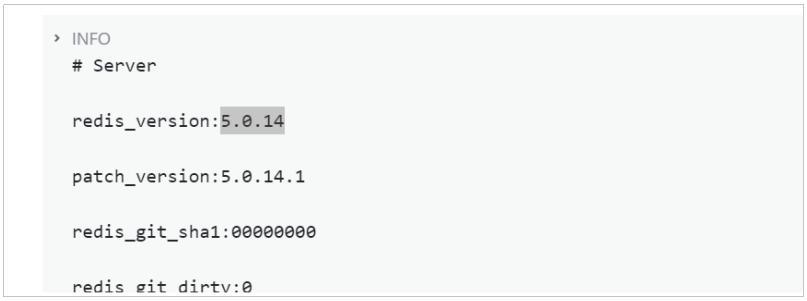
La edición básica de DCS for Redis se basa en Redis de código abierto. Redis de código abierto utiliza un único hilo principal para procesar comandos, por lo que solo se utiliza un núcleo de CPU en cada nodo de Redis.

Debido a esta restricción, puede usar un clúster y agregar particiones para lograr un mayor rendimiento de la CPU. A cada nodo en una instancia de clúster se le asigna al menos un núcleo de CPU por defecto.

1.8 ¿Cómo puedo ver la versión de una instancia de DCS Redis?

Conéctese a la instancia y ejecute el comando **INFO**.

Figura 1-4 Consulta de información de instancia



2 Cliente y conexión de red

2.1 ¿Cómo configurar un grupo de seguridad?

Las instancias DCS Redis 3.0/4.0/5.0/6.0 y Memcached se despliegan en diferentes modos. Por lo tanto, el método de control de acceso varía.

- Para controlar el acceso a las instancias de la edición profesional de DCS Redis 3.0, Memcached y Redis 6.0, puede utilizar grupos de seguridad. Las listas blancas no son compatibles. Las operaciones del grupo de seguridad se describen en esta sección.
- Para controlar el acceso a las instancias de DCS Redis 4.0/5.0/6.0 edición básica, puede utilizar listas blancas. No se admiten grupos de seguridad. Las operaciones de la lista blanca se describen en **Gestión de la lista blanca de direcciones IP**.

A continuación se describe cómo configurar los grupos de seguridad para **intra-VPC access** y **public access** para las instancias de la edición profesional de DCS Redis 3.0, Memcached y Redis 6.0.

Acceso dentro de la VPC a las instancias de la edición profesional de DCS Redis 3.0, Memcached y Redis 6.0

Un ECS puede comunicarse con una instancia de DCS si pertenecen a la misma VPC y las mismas reglas de subred y grupo de seguridad están configuradas correctamente.

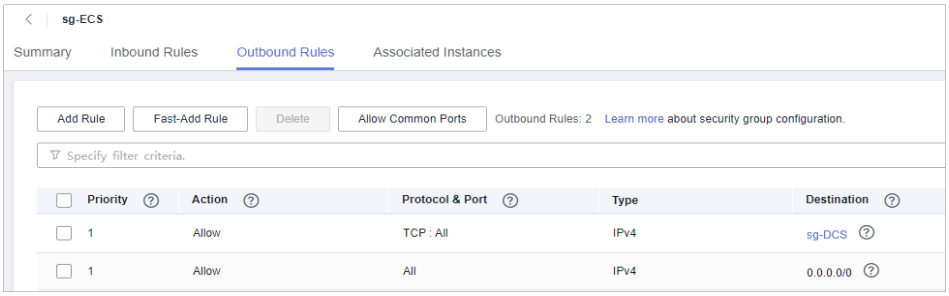
- Si la instancia de ECS y de DCS están configuradas con el mismo grupo de seguridad, el acceso a la red en el grupo no está restringido de forma predeterminada.
- Si la instancia de ECS y de DCS están configuradas con diferentes grupos de seguridad, agregue reglas de grupo de seguridad para asegurarse de que la instancia de ECS y de DCS puedan tener acceso entre sí.

  NOTA

  - Se supone que el ECS en el que se ejecuta el cliente pertenece al grupo de seguridad **sg-ECS** y que la instancia de DCS a la que accederá el cliente pertenece al grupo de seguridad **sg-DCS**.
  - A continuación se utiliza el puerto 6379 para instancias de DCS Redis 3.0 como ejemplo. Para otros casos, utilice el puerto real.
  - El extremo remoto es un grupo de seguridad o una dirección IP.

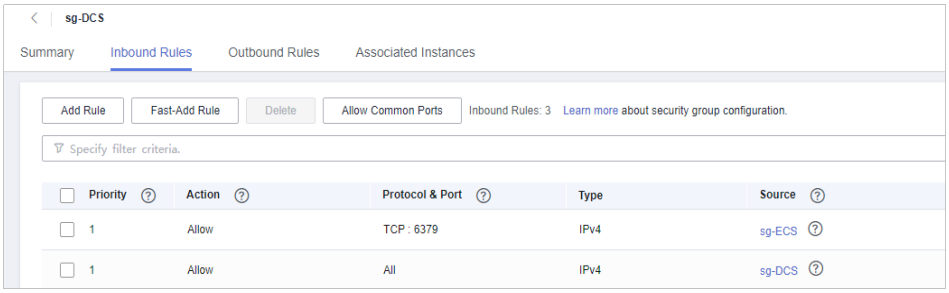
a. Configuración del grupo de seguridad para el ECS.

Agregue la siguiente regla de salida para permitir que el ECS acceda a la instancia de DCS. Omita esta regla si no hay restricciones en el tráfico saliente.



b. Configuración del grupo de seguridad para la instancia de DCS.

Para asegurarse de que el cliente puede acceder a la instancia de DCS, agregue la siguiente regla de entrada al grupo de seguridad configurado para la instancia de DCS:



AVISO

Para la dirección IP de origen, utilice la dirección IP especificada de la instancia de DCS. Evite usar **0.0.0.0/0** para evitar que los ECS vinculados al mismo grupo de seguridad sean atacados por vulnerabilidades de Redis.

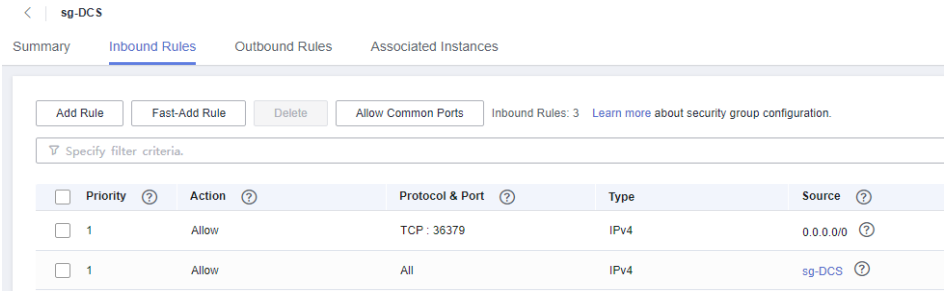
Acceso público a las instancias de DCS Redis 3.0

Un cliente puede acceder a una instancia de DCS solo después de que las reglas estén correctamente configuradas para el grupo de seguridad de la instancia.

Por ejemplo, para **sg-DCS** de grupo de seguridad, debe configurar las siguientes reglas en la dirección entrante:

Establezca el protocolo en TCP y la dirección IP de origen en 0.0.0.0/0 o una dirección de cliente especificada. Si SSL está habilitado, establezca el número de puerto en 36379. Si SSL está deshabilitado, establezca el número de puerto en 6379. Consulta la siguiente figura.

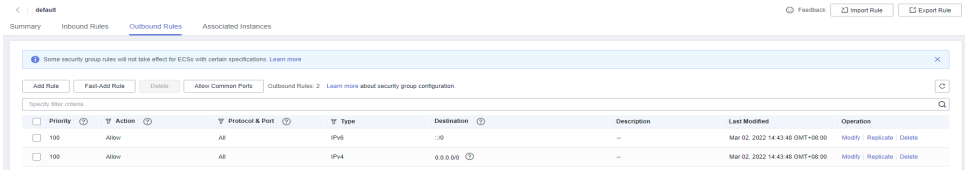
Figura 2-1 Regla de grupo de seguridad (para el ejemplo se utiliza el puerto 36379)



Grupo de seguridad de una tarea de migración

- Cuando cree una tarea de migración en línea, seleccione un grupo de seguridad. Sus reglas de salida deben permitir el tráfico a través de las direcciones IP y los puertos de las instancias de origen y destino de Redis. De forma predeterminada, se permite todo el tráfico saliente.
- La importación de copias de respaldo utiliza el grupo de seguridad **default**. Asegúrese de que todo el tráfico saliente está permitido (esta es la configuración predeterminada)

Figura 2-2 Reglas de salida del grupo de seguridad de migración



2.2 ¿DCS apoya el acceso público?

- Redis 3.0

  En la actualidad, el **acceso público solo es compatible con las instancias de DCS Redis 3.0 protegidas con contraseña**. Puede habilitar o deshabilitar SSL para el acceso público. Se recomienda descargar un certificado de CA por adelantado y usarlo para verificar el certificado de una instancia de DCS por motivos de seguridad. Para obtener más información, consulte **Acceso público a una instancia de DCS Redis**.

- Redis 4.0 y Redis 5.0

  Las instancias de DCS Redis 4.0, 5.0 y 6.0 no admiten el acceso público. Si se requiere el acceso público para una instancia de nodo único, principal/en espera o de Clúster Proxy, utilice Nginx para redirigir conexiones a través de un ECS configurado con la misma VPC y grupo de seguridad que la instancia de DCS. Para obtener más información, consulte la sección **Uso de Nginx para el acceso público a la instancia de nodo único, principal/en espera o de Clúster Proxy de DCS Redis**.

  No se puede acceder a las instancias de Clúster Redis usando Nginx a través de las redes públicas.

  Puede utilizar Elastic Load Balance (ELB) para acceder a los tipos diferentes de instancias de DCS a través de las redes públicas. Para obtener más información, consulte **Uso de ELB para el acceso público a DCS**.

- Memcached

No se admite el acceso público. El ECS que sirve como cliente y la instancia de DCS a la que accederá el cliente deben pertenecer a la misma VPC. En las fases de desarrollo y depuración de aplicaciones, también puede usar SSH para acceder a su instancia en el entorno local. Para obtener más información, consulte **Uso de túnel SSH para el acceso público a una instancia de DCS**.

2.3 ¿DCS admite el acceso entre las VPC?

Entre las VPC significa que el cliente y la instancia no están en la misma VPC.

A continuación se supone que el acceso público está deshabilitado para una instancia de DCS. Generalmente, las VPC se aíslan entre sí y un ECS no puede acceder a una instancia de DCS que pertenece a una VPC diferente del ECS.

Sin embargo, al establecer interconexiones de VPC entre las VPC, un ECS puede acceder a una instancia de DCS entre las VPC.

Cuando utilice las interconexiones de VPC para acceder a instancias de DCS en VPC, siga las siguientes reglas:

- Si se utilizan los bloques CIDR 172.16.0.0/12 a 172.16.0.0/24 durante la creación de instancias de DCS, el cliente no puede estar en ninguno de los siguientes bloques CIDR: 192.168.1.0/24, 192.168.2.0/24 y 192.168.3.0/24.
- Si se utilizan bloques CIDR 192.168.0.0/16 a 192.168.0.0/24 durante la creación de instancias de DCS, el cliente no puede estar en ninguno de los siguientes bloques CIDR: 172.31.1.0/24, 172.31.2.0/24 y 172.31.3.0/24.
- Si se utilizan bloques CIDR 10.0.0.0/8 a 10.0.0.0/24 durante la creación de instancias de DCS, el cliente no puede estar en ninguno de los siguientes bloques CIDR: 172.31.1.0/24, 172.31.2.0/24 y 172.31.3.0/24.

Para obtener más información sobre cómo crear y usar interconexiones de VPC, consulte la **Interconexión de VPC**.

2.4 ¿Se me cobrará por la EIP utilizada para el acceso público a una instancia de DCS Redis?

Sí. Debe pagar por la EIP utilizada para el acceso público a una instancia de DCS Redis 3.0.

Antes de habilitar el acceso público, debe tener una EIP disponible. Para ver los detalles de facturación, consulte los **detalles de precios de EIP**.

2.5 ¿Por qué se muestra "(error) NOAUTH Authentication required" cuando accedo a una instancia de DCS Redis?

Esto se debe a que ha habilitado el acceso sin contraseña para la instancia. Para evitar que aparezca el mensaje de error, no introduzca ninguna contraseña.

2.6 ¿Qué debo hacer si el acceso a DCS falla después de que el servidor se desconecta?

Análisis: Si se utilizan conexiones persistentes ("pconnect" en la terminología de Redis) o agrupación de conexiones y las conexiones se cierran después de ser utilizadas para conectarse a instancias de DCS, se devolverán errores al intentar reutilizar las conexiones.

Solución: cuando utilice pconnect o agrupación de conexiones, no cierre la conexión después del final de una solicitud. Si la conexión se interrumpe, restablezca de nuevo.

2.7 ¿Por qué las solicitudes a veces se agotan en los clientes?

Los errores de tiempo de espera ocasionales son normales debido a la conectividad de red y las configuraciones de tiempo de espera del cliente.

Se recomienda incluir operaciones de reconexión en su código de servicio para evitar errores en el servicio si falla una sola solicitud.

Si se agota el tiempo de espera de una solicitud de conexión, compruebe si se ha habilitado la persistencia de AOF. Para evitar el bloqueo, asegúrese de que se ha habilitado AOF.

Si se producen los errores de tiempo de espera con frecuencia, póngase en contacto con el soporte técnico.

2.8 ¿Qué debo hacer si se devuelve un error cuando uso el grupo de conexiones de Jedis?

El mensaje de error que posiblemente se mostrará cuando use el grupo de conexión de Jedis JedisPool es el siguiente:

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

Si se muestra este mensaje de error, compruebe si la instancia se está ejecutando correctamente. Si se está ejecutando correctamente, realice las siguientes comprobaciones:

Paso 1 Comprobar el estado de la red.

1. Compruebe las configuraciones de la dirección IP.

   Compruebe si la dirección IP configurada en el cliente Jedis es la misma que la dirección de subred configurada para su instancia de DCS. Si el acceso público está habilitado para su instancia, compruebe si la dirección IP configurada en el cliente de Jedis es la misma que la EIP vinculada a su instancia. Si no son coherentes, modifique la configuración de la dirección IP e inténtelo de nuevo.

2. Pruebe la red.

   Utilice el comando ping y telnet en el cliente para probar la red.

   – Si no se puede hacer ping a la red:

- Para el acceso dentro de la VPC, asegúrese de que el cliente y su instancia de DCS están en la misma VPC, y **las reglas de grupo de seguridad o la lista blanca** se han configurado correctamente.
- Para el acceso público con SSL, asegúrese de haber configurado el grupo de seguridad de su instancia de DCS, permitiendo el acceso a través del puerto 36379 como se indica en **Configuraciones de grupo de seguridad**.
- Para el acceso público sin SSL, asegúrese de haber configurado el grupo de seguridad de su instancia de DCS, permitiendo el acceso a través del puerto 6379 como se indica en **Configuraciones de grupo de seguridad**.

– Si la dirección IP se puede hacer ping pero telnet falló, reinicie su instancia. Si el problema persiste después del reinicio, póngase en contacto con el soporte técnico.

Paso 2 Compruebe el número de conexiones.

Compruebe si el número de conexiones de red establecidas excede el límite superior configurado para JedisPool. Si el número de conexiones establecidas se acerca al límite superior configurado, reinicie el servicio de DCS y compruebe si el problema persiste. Si el número de conexiones establecidas está muy por debajo del límite superior, continúe con las siguientes comprobaciones.

En Unix o Linux, ejecute el siguiente comando para consultar el número de conexiones de red establecidas:

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

En Windows, ejecute el siguiente comando para consultar el número de conexiones de red establecidas:

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

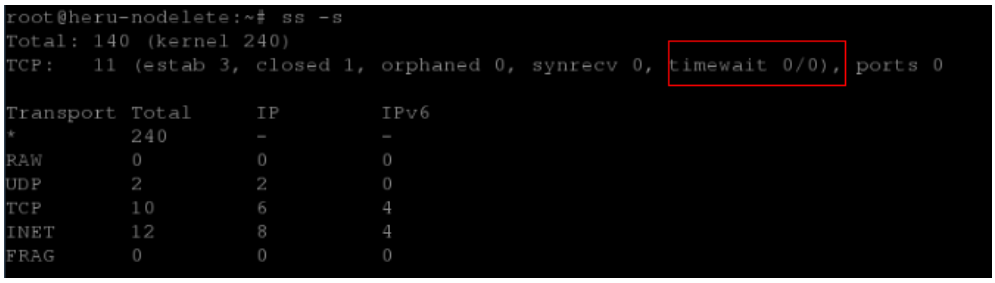
Paso 3 Compruebe el código de JedisPool.

Si el número de conexiones establecidas se aproxima al límite superior, determine si el problema se debe a la concurrencia del servicio o al uso incorrecto de JedisPool.

Al usar JedisPool, debe invocar a **jedisPool.returnResource()** o **jedis.close()** (recomendado) para liberar los recursos después de invocar a **jedisPool.getResource()**.

Paso 4 Compruebe el número de conexiones TIME_WAIT.

Ejecute el comando **ss -s** para comprobar si hay demasiadas conexiones **TIME_WAIT** en el cliente.



Si hay demasiadas conexiones **TIME_WAIT**, modifique los parámetros del núcleo ejecutando el comando **/etc/sysctl.conf** de la siguiente manera:

```
##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
```

Después de la modificación, ejecute el comando **/sbin/sysctl -p** para que la modificación surta efecto.

Paso 5 Si el problema persiste después de realizar las comprobaciones anteriores, realice los siguientes pasos.

Capture paquetes y envíe archivos de paquetes junto con la hora y la descripción de la excepción al soporte técnico para su análisis.

Ejecute el siguiente comando para capturar paquetes:

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

En Windows, también puede instalar la herramienta de Wireshark para capturar paquetes.

NOTA

Para el acceso público, cambie el número de puerto a **36379**.

Reemplace el nombre de la NIC por el nombre real.

----Fin

2.9 ¿Cómo puedo acceder a una instancia de DCS Redis a través de Redis Desktop Manager?

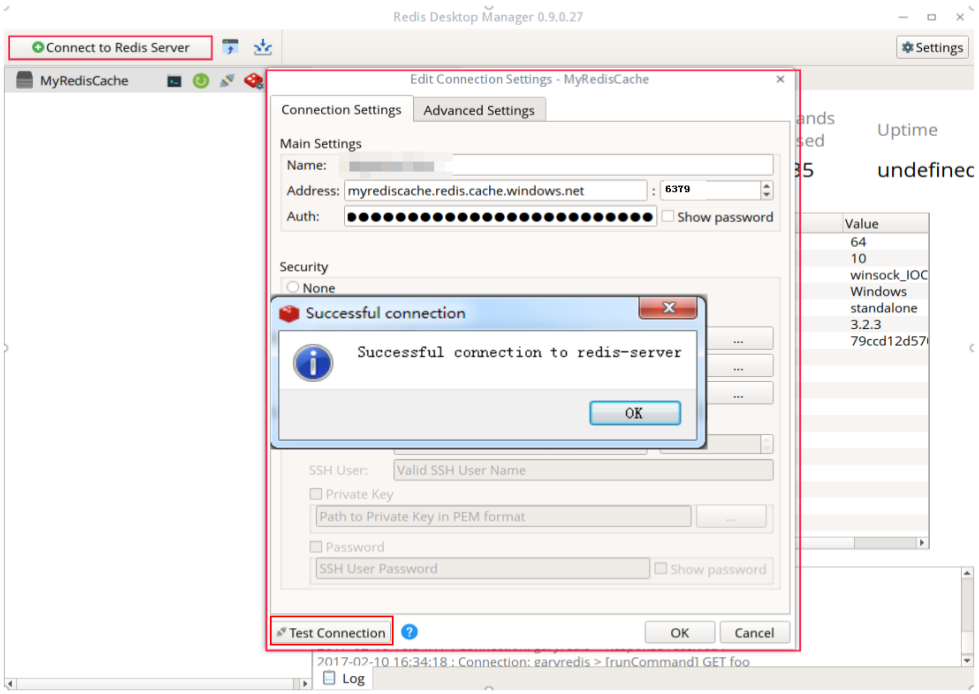
Puede acceder a una instancia de DCS Redis 3.0 a través de Redis Desktop Manager dentro de una VPC o a través de Internet.

Dentro de una VPC

1. Introduzca la dirección, el número de puerto (6379) y la contraseña de autenticación de la instancia de DCS a la que desea acceder.
2. Haga clic en **Test Connection**.

   El sistema muestra un mensaje de éxito si la conexión se realiza correctamente.

Figura 2-3 Acceso a una instancia de DCS Redis a través de Redis Desktop Manager a través de la intranet



NOTA

Al acceder a una instancia DCS de clúster, el comando Redis se ejecuta correctamente, pero puede aparecer un mensaje de error a la izquierda porque los clústeres de DCS se basan en Codis, que difiere del Redis nativo en términos de la salida del comando **INFO**.

Por el Internet

Compruebe si SSL está habilitado para la instancia de DCS a la que desea acceder.

- Si SSL no está habilitado, introduzca la dirección de acceso público de la instancia.

  Configure la regla de entrada del grupo de seguridad de la instancia, permitiendo el acceso a través del puerto 6379.

- Si SSL está habilitado, instale el cliente de Stunnel y, a continuación, conéctese al servidor de Redis a través de Redis Desktop Manager. Notas:
  - El cliente de Stunnel debe estar instalado. Para obtener más información sobre cómo instalar y configurar el cliente de Stunnel, consulte **Instrucciones de Stunnel**.
  - La dirección debe establecerse en **127.0.0.1** en lugar de en la dirección IP pública. De lo contrario, se devolverá el "connection reset".

    Cuando SSL está habilitado, se accede a Redis a través de un canal cifrado establecido por Stunnel. Después de enviar una solicitud desde Redis Desktop Manager al puerto de escucha de 127.0.0.1, la solicitud se cifra y se envía a la instancia de Redis a través del puerto 36379 a través de una red pública.

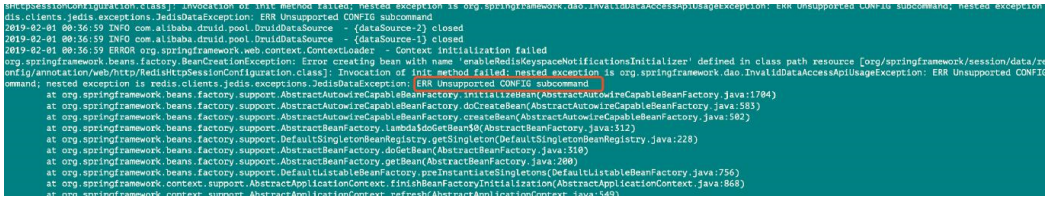
    Configure la regla de entrada del grupo de seguridad de la instancia, permitiendo el acceso a través del puerto 36379.

Para habilitar SSL, deshabilite primero el acceso público. A continuación, habilite SSL mientras vuelve a habilitar el acceso público. Para deshabilitar SSL, deshabilite primero el acceso público. A continuación, deshabilite SSL mientras vuelve a habilitar el acceso público.

2.10 ¿Qué sucede si SpringCloud muestra "ERR Unsupported CONFIG subcommand"?

Mediante el uso de instancias de DCS Redis, Spring Session puede desplegar el uso compartido de sesiones. Cuando se interconecta con Spring Cloud, se muestra la siguiente información de error:

Figura 2-4 Información de error de Spring Cloud



Por motivos de seguridad, DCS no admite el comando **CONFIG** iniciado por un cliente. Debe realizar los siguientes pasos:

1. En la consola de DCS, establezca el valor del parámetro **notify-keyspace-event** en **Egx** para una instancia de DCS Redis.
2. Agregue el siguiente contenido al archivo de configuración de XML del marco de Spring:

   <util:constant

   static-field="org.springframework.session.data.redis.config.ConfigureRedisAction.NO_OP"/>
3. Modifique el código de Spring relacionado. Habilite el componente bean **ConfigureRedisAction.NO_OP** para prohibir que un cliente invoque el comando **CONFIG**.

   @Bean

   public static ConfigureRedisAction configureRedisAction() {

   return ConfigureRedisAction.NO_OP;

   }

Para obtener más información, consulte la **Documentación de Spring Session**.

AVISO

El uso compartido de sesiones solo es compatible con las instancias de DCS Redis de **nodo único** y **principal/en espera** pero no con las instancias de clúster de DCS Redis.

2.11 ¿Qué puedo hacer si no puedo acceder a una instancia de DCS usando su dirección de nombre de dominio?

Si un cliente no puede conectarse a una instancia de DCS usando la dirección de nombre de dominio, establezca la dirección del servidor de DNS de la subred en la dirección privada del servidor de DNS.

Para obtener más información, consulte **¿Cómo cambio a un servidor privado de DNS?**

2.12 ¿Es necesaria una contraseña para acceder a una instancia? ¿Cómo configuro una contraseña?

- Se puede acceder a una instancia de DCS Redis con o sin contraseña. Puede acceder directamente a una instancia de DCS Redis a través de un cliente de Redis sin establecer una contraseña. Sin embargo, por motivos de seguridad, se recomienda establecer una contraseña para la autenticación y verificación siempre que sea posible. La contraseña se debe establecer al crear la instancia.
- Se puede acceder a una instancia de DCS Memcached con o sin contraseña. Puede seleccionar cualquier cliente de Memcached que admita el protocolo de texto y el protocolo binario de Memcached basándose en características específicas de la aplicación. La contraseña se debe establecer al crear la instancia.
- Para cambiar el modo de acceso a la instancia de Redis, o cambiar o restablecer una contraseña, consulte la sección **Gestión de contraseñas**.

2.13 ¿Puedo acceder a instancias de DCS en un entorno local?

- Si el acceso público está deshabilitado para una instancia de DCS, no puede acceder a ella en los entornos locales y solo puede acceder a ella a través de un ECS en una VPC que pueda comunicarse con la instancia. Las VPC se utilizan para garantizar la seguridad de la red de los servicios.

  Puede conectarse a una instancia de DCS desde su entorno local mediante un ECS que pueda comunicarse con su instancia para reenviar sus solicitudes. Para obtener más información, consulte **Uso de túnel SSH para el acceso público a una instancia de DCS**.
- Si el acceso público está habilitado, se puede acceder a las instancias de DCS en los entornos locales. Para obtener más información, consulte **Acceso público a una instancia de DCS Redis**.

2.14 ¿Qué debe tenerse en cuenta al usar Redis para Pub/Sub?

El **sitio web oficial de Redis** describe Pub/Sub en detalle. Cuando utilice Redis para Pub/Sub, tenga en cuenta lo siguiente:

- Su cliente debe procesar los mensajes de manera oportuna.

  Su cliente se suscribe a un canal. Si no recibe mensajes de manera oportuna, los mensajes de instancia de DCS pueden estar sobrecargados. Si el tamaño de los mensajes acumulados alcanza el umbral (32 MB por defecto) o permanece en un cierto nivel (8 MB por defecto) durante un cierto período de tiempo (1 minuto por defecto), su cliente se desconectará automáticamente para evitar el agotamiento de la memoria del servidor.

- Su cliente debe admitir el restablecimiento de la conexión en caso de desconexión.

  En caso de desconexión, debe ejecutar el comando **subscribe** o **psubscribe** en su cliente para suscribirse de nuevo a un canal. De lo contrario, el cliente no puede recibir mensajes.

- No utilice pub/sub en escenarios con los requisitos altos de confiabilidad de mensajes.

  El Redis pub/sub no es un sistema de mensajería fiable. Los mensajes que no se recuperan se descartarán cuando el cliente esté desconectado o se produzca una conmutación principal/en espera.

2.15 ¿Por qué el acceso público a mi instancia de DCS Redis está deshabilitado involuntariamente?

Síntoma: Se ha habilitado el acceso público para una instancia de DCS Redis 3.0, pero se deshabilita de repente.

Causa: la EIP vinculada a la instancia de DCS Redis no está vinculada. Como resultado, el acceso público se deshabilita automáticamente.

2.16 ¿Qué puedo hacer si se devuelve el error "Cannot assign requested address" cuando accedo a Redis usando connect?

Síntomas

El mensaje de error "Cannot assign requested address" aparece cuando se accede a Redis mediante **connect**.

Análisis

Las aplicaciones que encuentran este error suelen usar php-fpm y phpredis. En escenarios de alta concurrencia, un gran número de conexiones TCP están en el estado TIME-WAIT. Como resultado, el cliente no puede asignar los nuevos puertos y se devolverá el mensaje de error.

Soluciones

- Solución 1: Utilice **pconnect** en lugar de **connect**.

  El uso de **pconnect** reduce el número de conexiones TCP y evita que las conexiones se restablezcan para cada solicitud y, por lo tanto, reduce la latencia.

  Cuando se utiliza **connect**, el código para conectarse a Redis es el siguiente:

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


Reemplace **connect** por **pconnect** y el código se convertirá en:

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

NOTA

- Reemplace los parámetros de conexión del ejemplo con los valores reales. *${Hostname}*, *${Port}* y *${Inst_Password}* son la dirección de conexión, el número de puerto y la contraseña de la instancia de Redis, respectivamente.
- phpredis debe ser v5.3.0 o posterior. Se recomienda utilizar este modo de inicialización **pconnect** para evitar los errores NOAUTH durante la desconexión.

- Solución 2: Modifique el parámetro **tcp_max_tw_buckets** del ECS donde se encuentra el cliente.

  En esta solución, se reutilizan los puertos utilizados por las conexiones TIME-WAIT. Sin embargo, si se produce la retransmisión entre el ECS y el servicio de backend, la conexión puede fallar. Por lo tanto, se recomienda la solución **pconnect**.

  a. Conéctese al ECS donde se encuentra el cliente

  b. Ejecute el siguiente comando para comprobar los parámetros **ip_local_port_range** y **tcp_max_tw_buckets**:

     sysctl net.ipv4.tcp_max_tw_buckets net.ipv4.ip_local_port_range

     La información que aparecerá en pantalla será similar a la información siguiente:

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

  c. Ejecute el siguiente comando para establecer el parámetro **tcp_max_tw_buckets** en un valor menor que el valor de **ip_local_port_range**:

     sysctl -w net.ipv4.tcp_max_tw_buckets=10000

Generalmente, se recomienda la solución 1. En los escenarios especiales (por ejemplo, el código de servicio implica demasiados componentes y es difícil de cambiar), la solución 2 se puede usar para cumplir con los requisitos de alta simultaneidad.

2.17 Selección del grupo de conexiones y configuración recomendada de parámetros de Jedis

Ventajas del grupo de conexión de Jedis

La comparación entre Lettuce y Jedis es la siguiente:

- Lettuce
  - Lettuce no realiza la detección keepalive de la conexión. Si existe una conexión anormal en el grupo de conexión, se reporta un error cuando se agota el tiempo de espera de las solicitudes.
  - Lettuce no implementa la validación del grupo de conexión como **testOnBorrow**. Como resultado, las conexiones no se pueden validar antes de ser utilizadas.
- Jedis
  - Jedis implementa la validación de grupo de conexión usando **testOnBorrow**, **testWhileIdle** y **testOnReturn**.

Si **testOnBorrow** está habilitado, la validación de la conexión se realiza cuando se toman las conexiones, lo que tiene la mayor fiabilidad pero afecta al rendimiento (la detección se realiza antes de cada solicitud de Redis).

- **testWhileIdle** se puede utilizar para detectar las conexiones inactivas. Si el umbral está ajustado correctamente, las conexiones anormales en el grupo de conexión se pueden eliminar a tiempo para evitar errores de servicio causados por las conexiones anormales.
- Si una conexión se vuelve anormal antes de la comprobación de la conexión inactiva, el servicio que utiliza la conexión puede informar de un error. Puede especificar el parámetro **timeBetweenEvictionRunsMillis** para controlar el intervalo de comprobación.

Por lo tanto, Jedis tiene las mejores capacidades de manejo y detección de excepciones y es más confiable que Lettuce en los escenarios donde hay excepciones de conexión y jitter de la red.

Configuración recomendada del parámetro del grupo de conexión de Jedis

Tabla 2-1 Configuración recomendada del parámetro del grupo de conexión de Jedis

Parámetro	Descripción	Configuración recomendada
maxTotal	Número máximo de conexiones	Establezca este parámetro en función del número de subprocesos HTTP del contenedor web y de las conexiones reservadas. Se supone que el parámetro **maxConnections** del Tomcat Connector está establecido en **150** y cada solicitud de HTTP puede enviar simultáneamente dos solicitudes a Redis, se recomienda establecer este parámetro en al menos 400 (150 x 2 + 100). **Limit**: el valor de **maxTotal** multiplicado por el número de nodos de cliente (contenedores de CCE o máquinas virtuales de servicio) debe ser menor que el número máximo de las conexiones permitidas para una sola instancia de DCS Redis. Por ejemplo, si **maxClients** de una instancia de DCS Redis principal/en espera es 10,000 y **maxTotal** de un cliente único es 500, el número máximo de clientes es 20.
maxIdle	Número máximo de conexiones inactiva	Establezca este parámetro en el valor de **maxTotal**.

Parámetro	Descripción	Configuración recomendada
minIdle	Número mínimo de conexiones inactiva	Por lo general, se recomienda establecer este parámetro en 1/X de **maxTotal**. Por ejemplo, el valor recomendado es **100**. En los escenarios sensibles al rendimiento, puede establecer este parámetro en el valor de **maxIdle** para evitar el impacto causado por los cambios frecuentes en la cantidad de conexión. Por ejemplo, establezca este parámetro en **400**.
maxWaitMillis	Tiempo máximo de espera para obtener una conexión, en milisegundos	El tiempo de espera máximo recomendado para obtener una conexión desde el grupo de conexión es el tiempo de espera máximo tolerable de un solo servicio menos el tiempo de espera para la ejecución del comando. Por ejemplo, si el tiempo máximo de espera HTTP tolerable es 15s y el tiempo de espera de las solicitudes de Redis es 10s, establezca este parámetro en 5s.
timeout	Tiempo de espera de ejecución de comandos, en milisegundos	Este parámetro indica el tiempo de espera máximo para ejecutar un comando Redis. Establezca este parámetro basado en la lógica del servicio. Por lo general, se recomienda establecer este tiempo de espera a más de 210 ms para garantizar la tolerancia a fallas de red. Para la lógica de detección especial o la detección de excepciones de entorno, puede ajustar este tiempo de espera a segundos.

Parámetro	Descripción	Configuración recomendada
minEvictableIdleTimeMillis	Tiempo de desahucio de conexión inactiva, en milisegundos. Si una conexión no se utiliza durante un período mayor que este, se liberará.	Si no desea que el sistema restablezca con frecuencia las conexiones desconectadas, establezca este parámetro en un valor grande (xx minutos) o establezca este parámetro en **–1** y compruebe las conexiones inactivas periódicamente.
timeBetweenEvictionRunsMillis	Intervalo para detectar las conexiones inactivas, en milisegundos	El valor se estima en función del número de conexiones inactivas en el sistema. Por ejemplo, si este intervalo se establece en 30s, el sistema detecta conexiones cada 30s. Si se detecta una conexión anormal dentro de los 30 segundos, se eliminará. Establezca este parámetro en función del número de conexiones. Si el número de conexiones es demasiado grande y este intervalo es demasiado corto, los recursos de solicitud se desperdiciarán. Si hay cientos de conexiones, se recomienda establecer este parámetro en 30s. El valor se puede ajustar dinámicamente en función de los requisitos del sistema.
testOnBorrow	Indica si se debe comprobar la validez de la conexión mediante el comando **ping** cuando se toman prestadas conexiones del grupo de recursos. Se eliminarán las conexiones no válidas.	Si su servicio es extremadamente sensible a las conexiones y el rendimiento es aceptable, puede establecer este parámetro en **True**. Por lo general, se recomienda establecer este parámetro en **False** para habilitar la detección de la conexión inactiva.

Parámetro	Descripción	Configuración recomendada
testWhileIdle	Indica si se debe utilizar el comando **ping** para supervisar la validez de la conexión durante la supervisión de los recursos inactivos. Las conexiones no válidas serán destruidas.	Verdadero (True)
testOnReturn	Indica si se debe comprobar la validez de la conexión mediante el comando **ping** al devolver conexiones al grupo de recursos. Se eliminarán las conexiones no válidas.	Falso (False)
maxAttempts	Número de reintentos de conexión cuando se utiliza JedisCluster	Valor recomendado: 3–5. Valor predeterminado: **5**. Establezca este parámetro en función de los intervalos máximos de tiempo de espera de las API de servicio y de una sola solicitud. El valor máximo es de **10**. Si el valor excede de **10**, el tiempo de procesamiento de una sola solicitud es demasiado largo, bloqueando otras solicitudes.

2.18 ¿Qué puedo hacer si un cliente de Lettuce 6.x es incompatible con mi instancia de DCS?

Síntomas

Cuando un cliente de Lettuce 6.x se conecta a una instancia de Clúster Proxy DCS Redis 4.x/5.x, se muestra el mensaje de error "NOAUTH Authentication required".

Figura 2-5 Ejemplo de mensaje de error

```
[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)
```

Análisis

En Lettuce 6.x y las versiones posteriores, el comando **HELLO** de RESP3 (introducido en Redis 6.x) se utiliza para determinar la adaptación de la versión. Las instancias de las

versiones anteriores que no admiten el comando **HELLO** pueden tener los problemas de compatibilidad. Para estos casos, puede especificar el modo RESP2 (compatible con las versiones 4 y 5 de Redis) en Lettuce.

Solución

Agregue el siguiente código para usar el protocolo RESP2 para acceder a 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 ¿Debo usar un nombre de dominio o una dirección IP para conectarme a una instancia de DCS Redis?

- Clúster Proxy y nodo único:

  Cada instancia tiene solo una dirección IP y una dirección de nombre de dominio. Las direcciones permanecen sin cambios antes y después de la conmutación principal/en espera. Puede utilizar cualquiera de las direcciones para conectarse a la instancia.

- Principal/en espera (la edición básica):

  Cada instancia tiene una dirección IP y dos direcciones de nombre de dominio. Una de las direcciones de nombre de dominio se utiliza solo para procesar las solicitudes de lectura. Las direcciones permanecen sin cambios después de la conmutación principal/en espera. Puede utilizar cualquier dirección para conectarse a la instancia.

  Cuando utilice una dirección de nombre de dominio, distinga entre las solicitudes de lectura y escritura. Si utiliza **Connection Address** o **IP Address**, las funciones no se verán afectadas. Si utiliza **Read-only Address**, solo se procesan las solicitudes de lectura. Se recomienda utilizar las instancias de separación de lectura/escritura si tiene los requisitos relativos.

- Redis 6.0 (la edición profesional)

  Utilice el nombre de dominio para la conexión. Puede haber varias direcciones IP o pueden cambiar.

- Clúster Redis:

Una instancia de Clúster Redis tiene múltiples pares de direcciones IP maestras y de réplicas y una dirección de nombre de dominio. Puede utilizar cualquier dirección para conectarse a la instancia.

El nodo conectado envía las solicitudes al nodo correcto. Todos los nodos del clúster pueden recibir solicitudes. **Configure todas las direcciones IP** para evitar los puntos únicos de falla.

NOTA

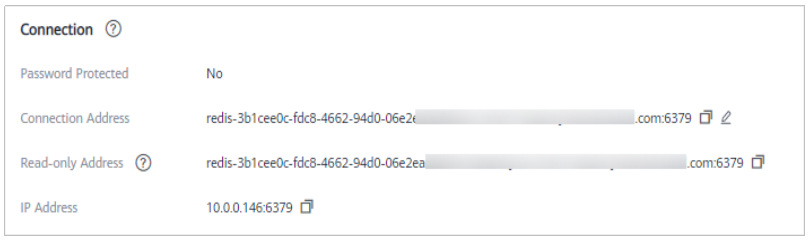
- Los nombres de dominio no se pueden resolver entre regiones. Si el cliente y la instancia de DCS Redis no están en la misma región, no se puede acceder a la instancia mediante su dirección de nombre de dominio. Puede asignar manualmente el nombre de dominio a la dirección IP del archivo **hosts** o acceder a la instancia mediante su dirección IP. Para obtener más información, véase **Restricciones**.
- Para obtener más información acerca de cómo conectarse a una instancia, consulte **Acceso a una instancia de DCS Redis**.

2.20 ¿La dirección de solo lectura de una instancia principal/en espera está conectada al nodo maestro o en espera?

Una instancia básica principal/en espera de DCS Redis 4.0/5.0/6.0 tiene una **dirección de conexión** y una **dirección de solo lectura**. La dirección de conexión se utiliza para conectarse al nodo principal de la instancia, y la dirección de solo lectura se utiliza para conectarse al nodo en espera de la instancia.

Para obtener más información, consulte la **Arquitectura de la instancia principal/en espera de la edición básica de DCS Redis 4.0/5.0/6.0**.

Figura 2-6 Direcciones de instancia



3 Uso de Redis

3.1 ¿Qué es la memoria reservada? ¿Cómo configuro la memoria reservada?

Memoria reservada

La memoria reservada es parte de la memoria que no se utiliza para almacenar datos, sino para la persistencia de datos, la sincronización principal/en espera y la copia de respaldo.

El parámetro **reserved-memory-percent** se utiliza para configurar la memoria reservada.

AVISO

En los datos de monitoreo, el uso de memoria no incluye el uso de memoria reservada.

Solo las instancias siguientes deben tener la memoria reservada:

- Instancias de DCS Redis 3.0 del nodo único
- Instancias principal/en espera de DCS Redis 3.0
- Instancias de DCS Memcached del nodo único
- Instancias principal/en espera de DCS Memcached

Si la memoria reservada es insuficiente porque los datos ocupan demasiada memoria, pueden producirse los siguientes problemas:

- Las operaciones en la instancia de DCS se vuelven lentas. (El sistema permite el intercambio, lo que deteriora el rendimiento.)
- No se puede realizar una copia de respaldo de los datos.
- Los datos no se pueden sincronizar entre los nodos principal y en espera a tiempo.
- Las especificaciones de instancia no se pueden cambiar.
- El proceso puede reiniciarse.

Procedimiento para configurar la memoria reservada

Cambie el valor de **reserved-memory-percent** haciendo referencia a **Modificación de parámetros de configuración de una instancia**.

NOTA

- Establezca el parámetro en **30** como mínimo. Para las instancias creadas en o después de 2021, el valor predeterminado es **30**.
- El porcentaje toma la memoria máxima disponible, en lugar de la memoria total, como el conjunto. La memoria disponible aparece en la columna **Available Memory** en **Especificaciones de instancias de DCS**.

3.2 ¿Qué son las cantidades de partición y de réplicas?

Partición

Una **partición** es una unidad de gestión en clústeres de Redis. Cada partición corresponde a un proceso de redis-servidor. Un clúster consta de varias particiones. Cada partición tiene varias ranuras. Los datos se almacenan de forma distribuida en las ranuras. Las particiones aumentan la capacidad de caché y las conexiones simultáneas.

Cada instancia de clúster consta de varias particiones. De forma predeterminada, cada partición es una instancia principal/en espera con dos réplicas. El número de particiones es igual al número de nodos principales en una instancia de clúster.

Réplica

Una réplica se refiere a un **nodo** de una instancia de DCS. Puede ser un nodo principal o un nodo en espera. Una instancia de réplica única no tiene nodo en espera. Una instancia de dos réplicas tiene un nodo principal y un nodo en espera. Por ejemplo, si el número de réplicas se establece en tres para una instancia principal/en standby, la instancia tiene un nodo principal y dos nodos en standby.

Número de réplicas y particiones de diferentes tipos de instancia

- **Nodo único**: Cada instancia tiene solo un nodo (un proceso Redis). Si el proceso de Redis es defectuoso, DCS inicia un nuevo proceso de Redis para la instancia.
- **Principal/en espera y separación de lectura/escritura**: Cada instancia tiene una partición que contiene un nodo principal y uno o más nodos en espera. Si el nodo principal está defectuoso, se activa la conmutación principal/en espera para restaurar los servicios. Cuantas más réplicas (nodos en espera), mejor será la confiabilidad (el rendimiento no se ve afectado).
- **Clúster**: Cada instancia tiene varias particiones. De forma predeterminada, cada partición es una instancia principal/en espera con dos réplicas. Por ejemplo, si una instancia de clúster tiene tres particiones y tres réplicas, cada partición tiene tres nodos (un nodo principal y dos nodos en espera).

Tipo de instancia	Particiones (Shards)	Réplicas	Balanceo de carga	Direcciones IP
Nodo único	1	-	-	1

Tipo de instancia	Particiones (Shards)	Réplicas	Balanceo de carga	Direcciones IP
Principal/en espera (la edición básica)	1	Predeterminado: 2; Personalizado: múltiple	No admitido	Igual que el número de réplicas
Principal/en espera (la edición profesional)	1	2 (no personalizado)	No admitido	Igual que el número de réplicas
Separación de lectura/escritura	1	Predeterminado: 2; Personalizado: múltiple	Admitido	1
Clúster Proxy	Múltiple	2 (no personalizado)	Admitido	1
Clúster Redis	Múltiple	Predeterminado: 2; Personalizado: uno o varios	No admitido	Número de réplicas x Número de partición

3.3 ¿Por qué el uso de CPU de una instancia de DCS Redis es 100%?

Síntomas

El uso de CPU de una instancia de Redis aumenta drásticamente en un corto período de tiempo. Si el uso de la CPU es demasiado alto, las conexiones pueden agotarse y se puede activar la conmutación principal/en espera.

Causas posibles

1. El servicio QPS es alto. En este caso, consulte **Comprobación de QPS**.
2. Se utilizaron los comandos que consumen recursos, como **KEYS**. En este caso, consulte **Localización y deshabilitación de comandos intensivos en la CPU**.
3. Se activó la reescritura de Redis. En este caso, consulte **Comprobación de reescritura de Redis**.

Comprobación de QPS

En la página **Cache Manager** de la consola de DCS, haga clic en una instancia para ir a la página de detalles de la instancia. En el menú de la izquierda, elija **Performance Monitoring** y, a continuación, vea la métrica **Ops per Second**.

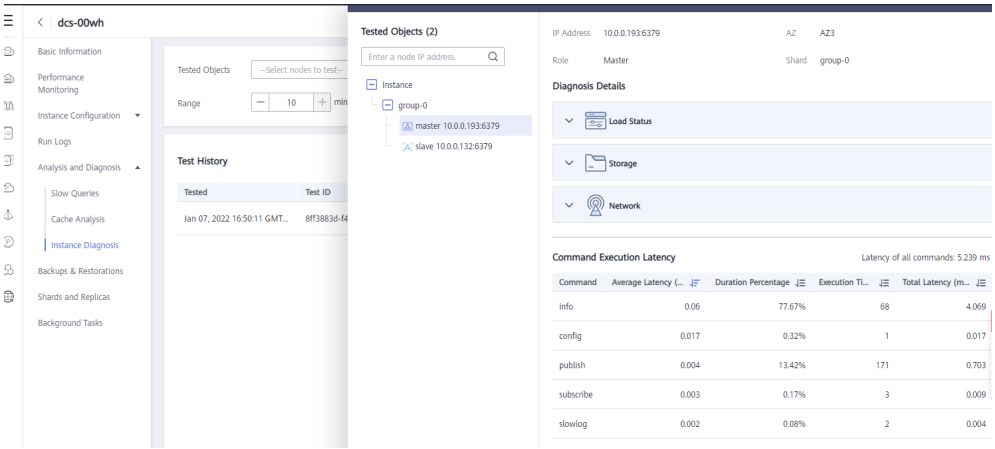
Localización y deshabilitación de comandos intensivos en la CPU

Se utilizan comandos que consumen recursos (comandos con complejidad de tiempo O(N) o superior), como **KEYS**. Generalmente, cuanto mayor es la complejidad temporal, más recursos utiliza un comando. Como resultado, el uso de la CPU es alto, y se puede activar fácilmente una conmutación principal/en espera. Para obtener más información sobre la complejidad temporal de cada comando, visite el **sitio web oficial de Redis**. En este caso, utilice el comando **SCAN** en su lugar o desactive el comando **KEYS**.

Paso 1 En la página **Performance Monitoring** de la consola de DCS, localice el período en el que el uso de la CPU es alto.

Paso 2 Utilice los métodos siguientes para encontrar los comandos que consumen un gran número de recursos.

- Redis registra las consultas que exceden una duración de ejecución especificada. Puede encontrar los comandos que consumen un gran número de recursos analizando las consultas lentas y su duración de ejecución. Para obtener más información, consulte la sección **Vista de consultas lentas de Redis**.
- Utilice la función de diagnóstico de instancia para analizar el porcentaje de duración de ejecución de diferentes comandos durante el período en que el uso de CPU es alto. Para obtener más información, consulte **Diagnóstico de una instancia**.



Paso 3 Resuelva el problema.

- Evalúe y deshabilite los comandos de alto riesgo y alto consumo, como **FLUSHALL**, **KEYS** y **HGETALL**.
- Optimice los servicios. Por ejemplo, evite las operaciones frecuentes de clasificación de datos.
- (Opcional) Realice las siguientes operaciones para ajustar las instancias en función de los requisitos de servicio:
  - Cambie el tipo de instancia a separación de lectura/escritura para separar las solicitudes de lectura y escritura de los comandos o aplicaciones de alto consumo.
  - Amplíe la instancia.

----Fin

Comprobación de reescritura de Redis

La persistencia de AOF, que está habilitada de forma predeterminada para las instancias principal/en espera y de clúster de DCS Redis, tiene lugar en los siguientes escenarios:

- Si se escribe una pequeña cantidad de datos y el archivo AOF no es grande, la reescritura de AOF se realiza de 01:00 a 04:00 de la mañana todos los días, y el uso de la CPU puede aumentar repentinamente durante este período.
- Cuando se escribe una gran cantidad de datos y el tamaño del archivo de AOF excede el umbral (de tres a cinco veces la capacidad de instancia de DCS), la reescritura de AOF se activa automáticamente en segundo plano independientemente de la hora actual.

La reescritura de Redis se realiza ejecutando el comando **BGSAVE** o **BGREWRITEAOF**, que puede consumir muchos recursos de CPU (vea **la discusión**). Los comandos **BGSAVE** y **BGREWRITEAOF** necesitan bifurcar(), lo que resulta en picos de uso de la CPU en un corto período de tiempo.

Si no se requiere persistencia, desactívela cambiando el valor de **appendonly** a **no** en la página **Parameters** de la instancia. Sin embargo, si desactiva la persistencia, la pérdida de datos puede ocurrir debido a la falta de vaciado de datos en el disco en las situaciones extremas.

3.4 ¿Puedo cambiar la VPC y la subred de una instancia de DCS Redis?

No. Una vez creada una instancia, su VPC y subred no se pueden cambiar. Si desea utilizar un conjunto diferente de VPC y subred, cree una misma instancia y especifique un conjunto deseado de VPC y subred. Una vez creada la nueva instancia, puede migrar datos de la instancia antigua a la nueva siguiendo las **instrucciones de migración de datos**.

3.5 ¿Por qué no se pueden configurar los grupos de seguridad para las instancias de edición básica de DCS Redis 4.0/5.0/6.0?

Actualmente, las instancias de edición básica de DCS Redis 4.0/5.0/6.0 usan los puntos de conexión de VPC y no admiten los grupos de seguridad.

Para permitir el acceso solo desde las direcciones IP específicas a una instancia de edición básica de DCS Redis 4.0, 5.0 o 6.0, agregue las direcciones IP a la lista blanca de la instancia.

Si no se agregan listas blancas para la instancia o la función de lista blanca está deshabilitada, todas las direcciones IP que pueden comunicarse con la VPC pueden acceder a la instancia.

Creación de un grupo de listas blancas

Paso 1 Inicie sesión en la **consola DCS**.

Paso 2 Haga clic en  en la esquina superior izquierda de la consola de gestión y seleccione una región.

NOTA

Seleccione la misma región que su servicio de aplicación.

Paso 3 En el panel de navegación, elija **Cache Manager**.

Paso 4 Haga clic en el nombre de una instancia de DCS.

Paso 5 Seleccione **Instance Configuration** > **Whitelist**. En la página mostrada, haga clic en **Create Whitelist Group**.

Paso 6 En el cuadro de diálogo **Create Whitelist Group**, especifique **Group Name** y **IP Address/Range**.

Tabla 3-1 Parámetros de la lista blanca

Parámetro	Descripción	Ejemplo
Group Name	Nombre del grupo de la lista blanca de la instancia. Se puede crear un máximo de cuatro grupos de listas blancas para cada instancia.	Prueba DCS
IP Address/ Range	Se puede agregar un máximo de 20 direcciones IP o intervalos de direcciones IP a una instancia. Separe las múltiples direcciones IP o los rangos de direcciones IP con comas. Direcciones IP y rango de direcciones IP no compatibles: 0.0.0.0 y 0.0.0/0.	10.10.10.1,10.10.10.10

Paso 7 Haz clic en **OK**.

Un grupo de lista blanca se habilita automáticamente para la instancia una vez creada. Solo las direcciones IP que se encuentren dentro de la lista blanca pueden acceder a la instancia.

NOTA

- En la lista de grupos de lista blanca, haga clic en **Edit** para modificar las direcciones IP o los intervalos de direcciones IP de un grupo y haga clic en **Delete** para eliminar un grupo de lista blanca.
- Después de habilitar la lista blanca, puede hacer clic en **Disable Whitelist** encima de la lista de grupos de listas blancas para permitir que todas las direcciones IP conectadas a la VPC accedan a la instancia.

----Fin

3.6 ¿Las instancias de DCS Redis limitan el tamaño de una clave o de un valor?

- El tamaño máximo permitido de una clave es de 512 MB.

  Para reducir el uso de memoria y facilitar la consulta de claves, asegúrese de que cada clave no exceda 1 KB.
- El tamaño máximo permitido de una cadena es de 512 MB.
- El tamaño máximo permitido de un Set, List, o Hash es de 512 MB.

  En esencia, un Set es una colección de cadenas; una List es una lista de cadenas; un Hash contiene asignaciones entre campos de cadena y valores de cadena.

Impedir que el cliente escriba constantemente los valores grandes en Redis. De lo contrario, se reducirá la eficiencia de la transmisión de la red y el servidor de Redis tardaría más tiempo en procesar los comandos, lo que resultaría en una mayor latencia.

3.7 ¿Puedo obtener las direcciones de los nodos en una instancia de DCS Redis de clúster?

Las instancias de Cluster DCS Redis 3.0 (tipo de Proxy Clúster) se utilizan de la misma manera que se utilizan las instancias de nodo único o principal/en espera. No es necesario conocer las direcciones del nodo de backend.

Para una instancia de clúster de DCS Redis 4.0 o 5.0 (tipo de Clúster Redis), ejecute el comando **CLUSTER NODES** para obtener las direcciones de nodo:

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

En la salida similar a la siguiente, obtenga las direcciones IP y los números de puerto de todos los nodos principales.

```
[root@ecs-[redacted]54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a [redacted]3 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
```

3.8 ¿Por qué la memoria disponible es más pequeña que el tamaño de caché de instancia?

Las instancias de DCS Redis 3.0 y de Memcached se despliegan en las máquinas virtuales, por lo que se reserva una pequeña cantidad de memoria para las sobrecargas del sistema. Este problema no se producirá en otras versiones de instancia.

3.9 ¿DCS for Redis admite la separación de lectura/escritura?

En la siguiente tabla se describe el soporte de DCS para la separación de lectura/escritura.

Tipo de instancia	Separación de lectura/escritura
Separación de lectura/escritura	Admitido. **NOTA** Para implementar la separación de lectura/escritura sin configuraciones de cliente, utilice **read/write splitting instances**.
Clúster Redis	La separación de lectura/escritura se puede configurar e implementar en el cliente. Para obtener más información, véase **Configuración**.
Principal/en espera (Redis 4.0/5.0/6.0)	La separación de lectura/escritura se puede implementar en un cliente que es capaz de distinguir entre solicitudes de lectura y escritura.
Otros	No se admite.

Configuración

- Para una **instancia de Clúster Redis**, puede consultar todos los nodos principales y de réplica ejecutando el comando **CLUSTER NODES**. El cliente se conectará a las réplicas y configurará el acceso de solo lectura en ellas.

  Ejecute el siguiente comando para consultar los nodos del clúster:

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

  La configuración de solo lectura en réplicas se logra mediante el comando **READONLY**.
- Para una **instancia básica principal/en espera de DCS Redis 4.0/5.0/6.0** hay dos nombres de dominio mostrados en la página de detalles de instancia de la consola: una dirección de lectura/escritura (nodo principal) y una dirección de solo lectura (nodo en espera). En el cliente, puede dirigir las solicitudes de escritura al nombre de dominio de lectura/escritura y las solicitudes de lectura al nombre de dominio de solo lectura.
- Para una **instancia de separación de lectura/escritura**, la separación de lectura/escritura se implementa en el lado del servidor de forma predeterminada. Los proxy distinguen entre las solicitudes de lectura y de escritura, y reenvían las solicitudes de escritura al nodo principal y las de lectura al nodo en espera. No es necesario realizar ninguna configuración en el cliente.

3.10 ¿DCS for Redis soporta multi-BD?

El soporte de DCS para múltiples bases de datos (multi-BD) es el siguiente:

- Instancias de DCS Redis de un solo nodo y principal/en espera: se admite multi-BD. Por defecto, hay 256 bases de datos, numerando 0–255. La base de datos predeterminada es DB0. Multi-BD se utiliza para el aislamiento de datos. El tamaño de cada base de datos no se asigna uniformemente. Como resultado, una base de datos puede ocupar completamente la memoria de la instancia.
- Clúster Proxy: Solo hay una base de datos por defecto.
  - Para obtener más información sobre cómo comprar una instancia de Clúster Proxy con múltiples bases de datos, consulte **¿Cómo Comprar una instancia de Clúster Proxy con multi-BD?**

  - Para obtener más información sobre cómo habilitar múltiples bases de datos para una instancia de Clúster Proxy con una sola base de datos, consulte **¿Cuáles son las limitaciones para desplegar múltiples bases de datos en una instancia de Clúster Proxy?**

    NOTA

    Las instancias de Clúster Proxy de DCS Redis 3.0 no soportan multi-BD.

- Instancias de Clúster Redis DCS: Multi-BD no es compatible. Solo hay una base de datos.

No se puede cambiar el número de bases de datos y no se puede personalizar el tamaño de cada base de datos.

3.11 ¿Cómo sé si una instancia es de BD única o de BD múltiples?

Nodo único, principal/en espera y separación de lectura/escritura: multi-BD (256 bases de datos, numeradas del 0 al 255)

Clúster Proxy: BD única por defecto. Multi-BD se puede activar. Para obtener más información, consulte **¿Cuáles son las limitaciones para implementar múltiples bases de datos en una instancia de Clúster Proxy?**

Clúster Redis: BD única. Multi-BD no es compatible.

Puede conectarse a una instancia de DCS Redis 4.0 o posterior en la consola para comprobar si es multi-BD.

Figura 3-1 Conexión a Redis

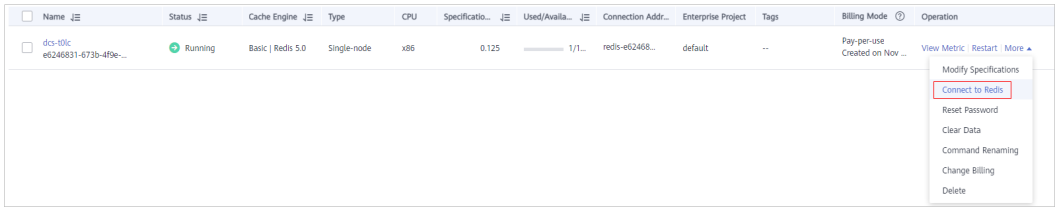
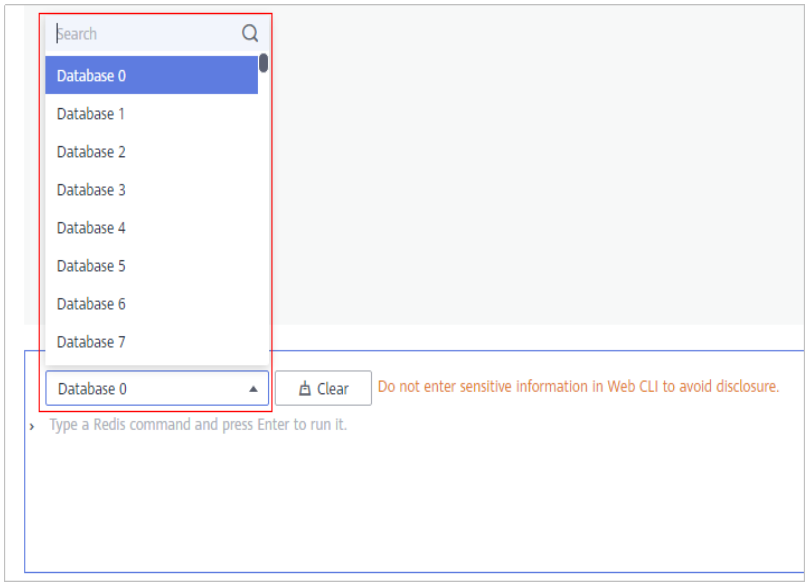


Figura 3-2 Consulta de bases de datos



3.12 ¿DCS for Redis admite Clúster Redis?

Sí. DCS for Redis 4.0 y 5.0 soporta Clúster Redis. DCS for Redis 3.0 admite Clúster Proxy y Clúster Redis.

3.13 ¿Qué es Sentinel?

Descripción general

La alta disponibilidad en Redis se despliega con Sentinel. Sentinel le ayuda a defenderse contra ciertos tipos de fallas sin intervención manual, y completa las tareas como monitoreo, notificación y configuración del cliente. Para obtener más información, visite el **sitio web oficial de Redis**.

principios

Redis Sentinel es un sistema distribuido donde varios procesos de Sentinel trabajan juntos. Ofrece los siguientes beneficios:

1. La detección de fallas solo se realiza cuando varios Sentinel aceptan que un nodo principal no está disponible, lo que reduce la posibilidad de falsos positivos.
2. Incluso si algunos procesos de Sentinel son defectuosos, el sistema de Sentinel aún puede funcionar correctamente para evitar fallas.

En un nivel superior, hay un sistema distribuido más grande que consiste en Sentinel, los nodos principales y de réplicas de Redis y los clientes conectados a Sentinel y Redis.

Características

- Monitoreo: Sentinel comprueba continuamente si los nodos principal y de réplica funcionan correctamente.

- Notificación: Si un nodo está defectuoso, Sentinel puede notificar al administrador del sistema u otros programas informáticos invocando a una API.
- Conmutación por error automática: Si el nodo principal es anormal, Sentinel inicia una conmutación por error para promover uno de réplica a principal. Otras réplicas replican datos del nuevo nodo principal. Las aplicaciones que utilizan la instancia para Redis recibirán una notificación de que deben conectarse a la nueva dirección.
- Configuración del cliente: Sentinel sirve como fuente autorizada para el descubrimiento del servicio al cliente. Los clientes se conectan a Sentinel y solicitan la dirección del nodo principal de Redis que es responsable de servicios específicos. Si se produce una conmutación por error, Sentinel entrega la nueva dirección.

3.14 ¿DCS for Redis es compatible con Sentinels?

- Las instancias de clúster y las instancias principal/en espera de DCS Redis 4.0, 5.0 y 6.0 admiten Sentinels. Sentinels supervisan el estado de ejecución de los nodos principales y en espera de una instancia principal/en espera y cada partición de una instancia de clúster. Si el nodo principal se vuelve defectuoso, se realizará una migración por falla.

  Sentinel son invisibles para usted y solo se utilizan en el servicio.
- DCS for Redis 3.0 no soporta Redis Sentinel. En su lugar, utiliza keepalive para monitorear los nodos maestros y de réplica y para gestionar la migración por falla.

3.15 ¿Cuál es la política predeterminada de desalojo de datos?

Los datos se desalojan de la caché basándose en un límite de espacio definido por el usuario con el fin de hacer espacio para nuevos datos. Para obtener más información, visite el **sitio web oficial de Redis**. En las versiones actuales de DCS for Redis, puede seleccionar una política de desalojo que prefiera.

Puede cambiar la política de desalojo configurando el parámetro **maxmemory-policy**.

Cuando se llega a **maxmemory**, puede seleccionar una de las siguientes ocho políticas de desalojo:

- **noeviction**: Cuando se alcanza el límite de memoria, las instancias de DCS devuelven errores a los clientes y ya no procesan solicitudes de escritura y otras solicitudes que podrían resultar en más memoria que se usará. Sin embargo, pueden seguir procesándose **DEL** y algunas solicitudes de excepción más.
- **allkeys-lru**: Las instancias de DCS intentan desalojar primero las claves menos usadas recientemente, con el fin de dejar espacio para los nuevos datos.
- **volatile-lru**: Las instancias de DCS intentan desalojar primero las claves menos usadas recientemente con un conjunto de caducidad, para dejar espacio para nuevos datos.
- **allkeys-random**: Las instancias de DCS reciclan las claves aleatorias para que se puedan almacenar nuevos datos.
- **volatile-random**: Las instancias de DCS desalojan las claves aleatorias con un conjunto caducado, con el fin de dejar espacio para nuevos datos.
- **volatile-ttl**: Las instancias de DCS desalojan claves con un conjunto caducado e intentan desalojar claves con un tiempo de vida más corto (TTL) primero, con el fin de hacer espacio para nuevos datos.

- **allkeys-lfu**: Las instancias de DCS desalojan las claves menos usadas de todas las claves.
- **volatile-lfu**: Las instancias de DCS desalojan las claves menos utilizadas con un campo **expire** desde todas las claves.

NOTA

- Si no se puede reciclar ninguna clave, **volatile-lru**, **volatile-random** y **volatile-ttl** son lo mismo que **noeviction**. Para obtener más información, consulte la descripción de **noeviction**.
- La política de desalojo predeterminada es **volatile-lru** para las instancias de DCS Redis creadas en o después de julio de 2020. La política de desalojo predeterminada es **noeviction** para las instancias de DCS Redis creadas antes del de julio de 2020.

3.16 ¿Qué debo hacer si ocurre un error en redis_exporter?

Inicie redis_exporter con la CLI. Basado en la salida, verifique los errores y solucione los problemas en consecuencia.

```
[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{""}
```

3.17 ¿Cómo puedo proteger mis instancias de DCS Redis?

Redis es una de las tecnologías de caché de código abierto más potentes y ampliamente utilizadas. Sin embargo, el Redis de código abierto no tiene características de seguridad robustas propias. Es vulnerable a ataques maliciosos de Internet, lo que puede causar violaciones de datos.

Para proteger sus instancias de DCS Redis, considere seguir los siguientes consejos:

- Configuraciones de conexión de red
  a. Cifrar los datos confidenciales y desactivar el acceso público.

     Los datos confidenciales deben cifrarse antes de almacenarse. No habilite el acceso público a menos que se requiera lo contrario.

  b. Configurar las reglas de acceso para los grupos de seguridad.

     Los grupos de seguridad y las VPC están diseñados para proteger el acceso a la red. Permitir el acceso a través del menor número posible de puertos para evitar riesgos.

  c. Configurar los firewalls de ECS.

     Configurar las reglas de filtrado de firewall para el ECS donde se ejecuta el cliente.

  d. Establecer la contraseña de la instancia.

  e. Configurar una lista blanca.
- uso de redis-cli
  a. Hide the password.

     Problema: Si se utiliza la opción **-a <password>**, la contraseña puede aparecer cuando se ejecuta el comando **ps**.

Solución: modifique el código fuente de Redis. Ocultar la contraseña inmediatamente después de iniciar redis-cli invocando a la función principal.

b. Deshabilitar sudo en secuencias de comandos en ejecución.

Problema: Los parámetros para iniciar redis-cli contienen los patrones sensibles relacionados con la contraseña, que pueden aparecer cuando se ejecuta el comando **ps** y pueden registrarse.

Solución: Acceda a la instancia invocando a las API (o a través de redis-py en Python). No permita cambiar al usuario **dbuser** usando sudo en redis-cli.

3.18 ¿Por qué las instancias de Clúster Proxy de DCS Redis 3.0 no soportan el bloqueo distribuido de Redisson?

Redisson despliega el bloqueo y el desbloqueo en el siguiente proceso:

1. El bloqueo y el desbloqueo de redisson se despliegan ejecutando scripts de Lua.
2. Durante el bloqueo, los comandos **EXISTS**, **HSET**, **PEXPIRE**, **HEXISTS**, **HINCRBY**, **PEXPIRE** y **PTTL** deben ejecutarse en el script de Lua.
3. Durante el desbloqueo, los comandos **EXISTS**, **PUBLISH**, **HEXISTS**, **PEXIPRE** y **DEL** deben ejecutarse en el script de Lua.

En un clúster basado en proxy, el proxy procesa los comandos **PUBLISH** y **SUBSCRIBE** y reenvía las solicitudes al servidor de Redis. El comando **PUBLISH** no se puede ejecutar en el script de Lua.

Como resultado, las instancias de Clúster Proxy de DCS Redis 3.0 no admiten bloqueos distribuidos de Redisson. **Para usar Redisson, recurra a Redis 4.0 o 5.0 en su lugar**.

3.19 ¿Puedo personalizar o cambiar el puerto para acceder a una instancia de DCS?

No se puede personalizar ni cambiar el puerto para acceder a una instancia de DCS Redis 3.0 o Memcached. Puede personalizar y cambiar el puerto para acceder a una instancia de DCS Redis 4.0, 5.0 o 6.0.

- Redis 3.0

  Acceso dentro de la VPC: puerto 6379; acceso público sin SSL: puerto 6379; acceso público con SSL: puerto 36379.

- Memcached

  Utilice el puerto 11211 para el acceso dentro de la VPC. No se admite el acceso público.

- Redis 4.0/5.0/6.0

  Puede especificar un puerto (entre 1 y 65535) o utilizar el puerto predeterminado (6379) para acceder a una instancia de DCS Redis 4.0, 5.0 o 6.0. Si no se especifica ningún puerto, se utilizará el puerto predeterminado.

  Las instancias de DCS Redis 4.0/5.0/6.0 no admiten el acceso público.

Si la instancia y el cliente están en diferentes grupos de seguridad, debe configurar reglas de acceso para los grupos de seguridad, permitiendo el acceso a través del puerto especificado. Para obtener más información, véase **¿Cómo configurar un grupo de seguridad?**.

Personalización de un puerto

Al crear una instancia de DCS Redis 4.0, 5.0 o 6.0, puede introducir un número de puerto para **IP Address**. Si no especifica un puerto, se utiliza el puerto predeterminado 6379.

Cambio del puerto

Después de crear una instancia de DCS Redis 4.0, 5.0 o 6.0, puede cambiar su puerto.

1. En el panel de navegación de la consola de DCS, elija **Cache Manager**.
2. Haga clic en una instancia de DCS Redis.
3. En el área **Connection**, haga clic en  junto a **Connection Address**.

AVISO

Después de cambiar el puerto, todas las conexiones a la instancia de Redis se interrumpen y los servicios se conectan al puerto nuevo.

3.20 ¿Puedo modificar las direcciones de conexión para acceder a una instancia de DCS?

Después de crear una instancia de DCS, su dirección IP y el nombre de dominio para el acceso dentro de la VPC no se pueden modificar. Si se ha habilitado el acceso público para la instancia, se puede modificar la dirección IP elástica (EIP) vinculada a la instancia.

Para utilizar una dirección IP diferente, debe crear una nueva instancia y especificar manualmente una dirección IP. Una vez creada la instancia, migre los datos de la instancia antigua a la nueva.

3.21 ¿Por qué no puedo eliminar una instancia?

Posibles causas y soluciones:

- La instancia no está en el estado **Running**.

  Solo se pueden eliminar instancias en el estado **Running**.
- Compruebe si no se puede crear la instancia.

  Para eliminar instancias que no se han creado, haga clic en el número junto a **Instance Creation Failures** en la consola de DCS.

3.22 ¿DCS admite el despliegue entre las AZ?

Las instancias principal/en espera, de separación de lectura/escritura y de clúster de DCS Redis y las instancias principal/en espera de DCS Memcached pueden desplegarse en todas las zonas de disponibilidad (AZ).

- Si los nodos de instancia de una AZ son defectuosos, los nodos de otras AZ no se verán afectados. El nodo en espera se convierte automáticamente en el nodo principal para continuar funcionando, lo que garantiza la recuperación ante desastres (DR).

- El despliegue entre las AZ no compromete la velocidad de sincronización de datos entre los nodos principal y en espera.

3.23 ¿Por qué se necesita mucho tiempo para iniciar una instancia de clúster de DCS?

Causa posible: cuando se inicia una instancia de clúster, el estado y los datos se sincronizan entre los nodos de la instancia. Si una gran cantidad de datos se escribe continuamente en la instancia antes de que se complete la sincronización, la sincronización se prolongará y la instancia permanecerá en el estado **Starting**. Una vez completada la sincronización, la instancia entra en el estado **Running**.

Solución: Comience a escribir datos en una instancia solo después de que la instancia se haya iniciado.

3.24 ¿DCS for Redis proporciona software de gestión de backend?

No. Para consultar la configuración y la información de uso de Redis, utilice redis-cli. Si desea supervisar las métricas de instancia de DCS Redis, vaya a la consola de Cloud Eye o realice las siguientes operaciones.

Procedimiento

Paso 1 Inicie sesión en la **consola DCS**.

Paso 2 Haga clic en en la esquina superior izquierda de la consola de gestión y seleccione la región donde se encuentra la instancia.

Paso 3 En el panel de navegación, elija **Cache Manager**.

Paso 4 Haga clic en la instancia deseada.

Paso 5 Elija **Performance Monitoring**. Se muestran todas las métricas de supervisión de la instancia.

NOTA

También puede hacer clic en **View Metric** en la columna **Operation** de la página **Cache Manager**. Serás redirigido a la consola Cloud Eye. Las métricas que se muestran en la consola de Cloud Eye son las mismas que las que se muestran en la página **Performance Monitoring** de la consola de DCS.

----Fin

3.25 ¿Por qué se utiliza la memoria de una instancia de DCS Redis por pocas claves?

Posible causa: El búfer de salida puede haber ocupado una cantidad excesiva de memoria.

Solución: Después de conectarse a la instancia mediante redis-cli, ejecute el comando **redis-cli --bigkeys** para buscar las claves grandes. A continuación, ejecute el comando **info** para comprobar el tamaño del búfer de salida.

3.26 ¿Puedo recuperar datos eliminados de una instancia de DCS?

Si ha realizado una copia de respaldo de la instancia de DCS, puede restaurar sus datos desde la copia de respaldo. Sin embargo, la restauración sobrescribirá los datos escritos antes de la restauración.

Puede restaurar los datos de copia de respaldo en una instancia principal/en espera, una de clúster o una de separación de lectura/escritura a través de **Backups & Restorations** en la consola de DCS. Para obtener más información, consulte la sección **Restauración de una instancia de DCS**.

Si se elimina una instancia de DCS, también se eliminarán los datos de la instancia y su copia de respaldo. Antes de eliminar una instancia, puede descargar los archivos de copia de respaldo de la instancia para almacenamiento local permanente y también puede migrarlos a una nueva instancia si necesita restaurar los datos. Para obtener más información sobre cómo descargar los datos de copia de respaldo, consulte **¿Cómo puedo exportar datos de instancia de DCS Redis?**

3.27 ¿DCS for Redis admite la transmisión cifrada de SSL?

De forma predeterminada, SSL está deshabilitado para las instancias de edición básica de DCS Redis 6.0. Para habilitarlo, consulte .

Para el acceso público a instancias de DCS (soportadas solo por las instancias de DCS Redis 3.0), puede habilitar la encriptación TLS con Stunnel. Para obtener más información, consulte las **instrucciones sobre la instalación y configuración de Stunnel**. Cuando DCS aprovisiona instancias, la Certificate Chain (CA) especificada generará un certificado de servicio único para cada instancia. Al conectarse a una instancia, los clientes pueden utilizar los certificados raíz de CA descargados de la consola de gestión para autenticar el servidor de instancia y cifrar datos durante la transmisión.

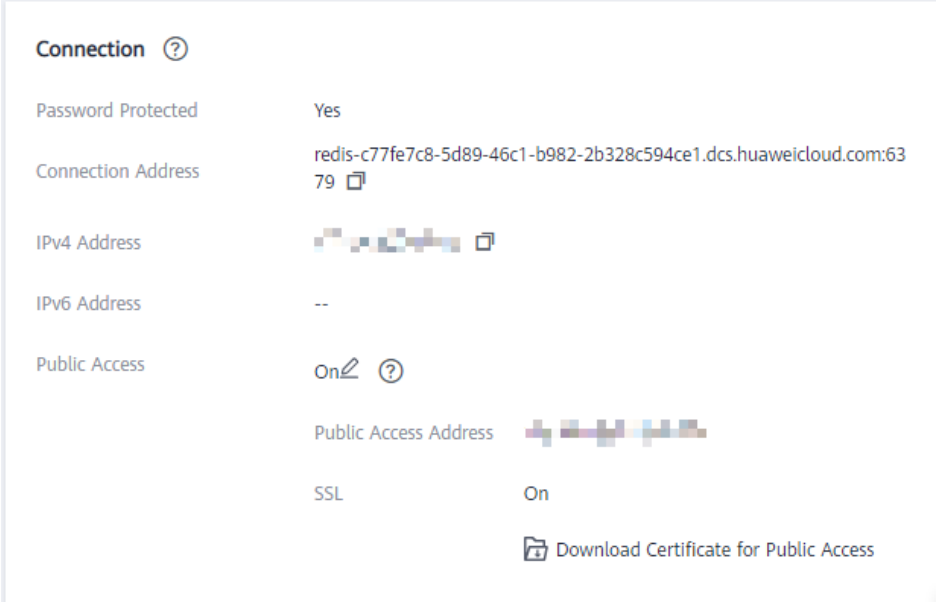
DCS Redis for 4.0 y 5.0 solo admite la transmisión de texto sin formato. No son compatibles con la transmisión cifrada de SSL.

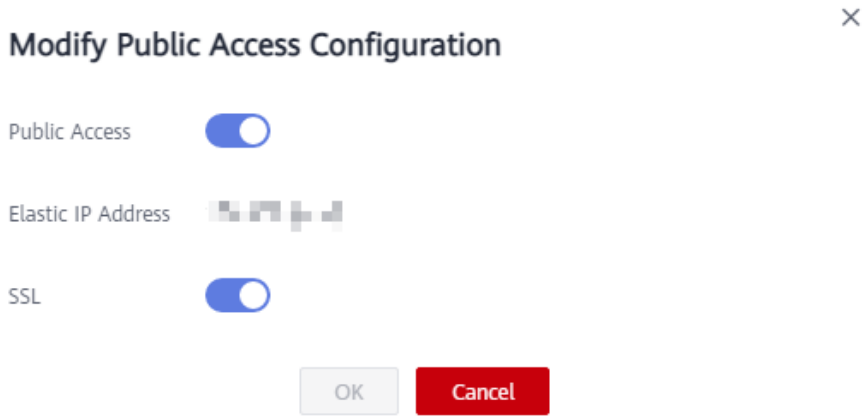
3.28 ¿Cómo puedo habilitar o deshabilitar SSL para el acceso público a una instancia de DCS Redis 3.0?

Cuando habilita el acceso público, SSL está habilitado de forma predeterminada.

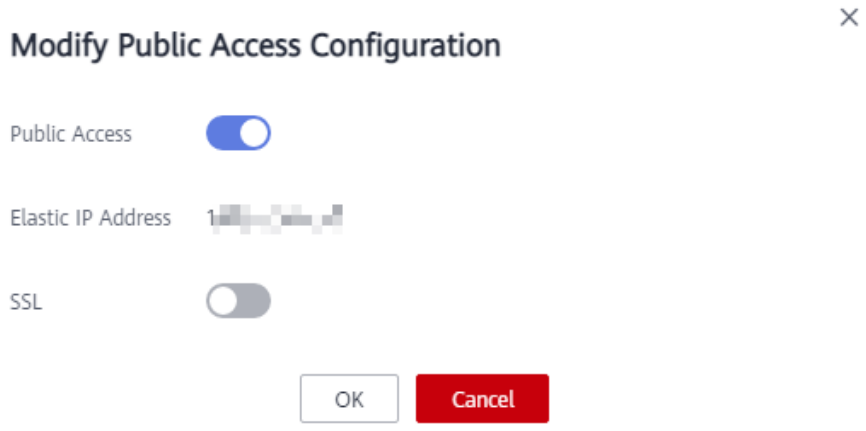
Para deshabilitar la encriptación de SSL, realice los siguientes pasos:

1. Abra la página para configurar el acceso público.





2. Deshabilite la encriptación SSL y haga clic en **OK**.



3. En el área **Connection** de la página de detalles de la instancia, **SSL** está deshabilitado.

3.29 ¿Por qué la memoria disponible de las instancias de DCS no utilizadas es menor que la memoria total y por qué el uso de la memoria de las instancias DCS no utilizadas es mayor que cero?

Para instancias de DCS Redis 3.0 y de Memcached, la memoria disponible es menor que la memoria total porque parte de la memoria está reservada para la sobrecarga del sistema y la persistencia de datos (soportada por las instancias principal/en espera). Las instancias de DCS utilizan una cierta cantidad de memoria para los búferes del servidor Redis y las estructuras de datos internas. Esta es la razón por la que el uso de memoria de las instancias de DCS no utilizadas es mayor que cero. Este problema no se producirá en otras versiones de instancia.

3.30 ¿Cómo calculo el uso de la memoria de Redis?

El uso estimado de memoria puede ser diferente del uso real de memoria. Actualmente, DCS for Redis proporciona las siguientes métricas relacionadas con la memoria:

Tabla 3-2 Métricas de instancias de DCS Redis 3.0

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
memory_usage	Uso de memoria	Consumo de memoria del objeto monitoreado Unidad: %	0–100%	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
used_memory	Memoria utilizada	Número de bytes utilizados por el servidor de Redis Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
used_memory_dataset	Conjunto de datos de memoria usado	Memoria de conjunto de datos que ha utilizado el servidor de Redis Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Compatible con Redis 4.0 y las versiones posteriores Dimensión: dcs_instance_id	1 minuto

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
used_memory_dataset_perc	Relación de conjunto de datos de memoria usado	Porcentaje de memoria de datos que Redis ha utilizado con respecto al total de memoria utilizada Unidad: %	0–100%	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Compatible con Redis 4.0 y las versiones posteriores Dimensión: dcs_instance_id	1 minuto
used_memory_rss	RSS de memoria usada	Memoria de tamaño de conjunto residente (RSS) que ha utilizado el servidor de Redis, que es la memoria que realmente reside en la memoria, incluida toda la memoria de pila y de montículo, pero no la memoria intercambiada Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
memory_frag_ratio	Relación de fragmentación de memoria	Fragmentación de memoria actual, que es la relación entre **used_memory_rss**/ **used_memory**.	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
used_memory_peak	Cantidad máxima de memoria usada	Memoria máxima consumida por Redis desde la última vez que se inició el servidor de Redis Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
used_memory_lua	Uso de memoria de Lua	Número de bytes utilizados por el motor Lua Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto

Tabla 3-3 Métricas de instancia de DCS Redis 4.0 y 5.0

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
memory_usage	Uso de memoria	Consumo de memoria del objeto monitoreado Unidad: %	0–100%	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
used_memory	Memoria utilizada	Número de bytes utilizados por el servidor de Redis Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
used_memory_dataset	Conjunto de datos de memoria usado	Memoria de conjunto de datos que ha utilizado el servidor de Redis Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
memory_frag_ratio	Relación de fragmentación de memoria	Relación entre la memoria usada RSS y la memoria usada	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
used_memory_lua	Uso de memoria de Lua	Número de bytes utilizados por el motor Lua Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto
used_memory_peak	Cantidad máxima de memoria usada	Memoria máxima consumida por Redis desde la última vez que se inició el servidor de Redis Unidad: byte	≥ 0	Objeto supervisado: Instancia de DCS Redis de nodo único, principal/en espera o de clúster Dimensión: dcs_instance_id	1 minuto

3.31 La capacidad y el rendimiento de la instancia de Clúster Redis están todavía bajos, ¿por qué se sobrecarga la capacidad o el rendimiento de una partición?

Clúster Redis utiliza un método especial de la partición de datos. **Cada clave es parte de una ranura de hash, que es mantenida por un nodo en el clúster.** Para calcular cuál es la ranura de hash de una clave dada:

1. Tome el CRC16 del módulo de clave 16384.
2. En base a la asignación entre ranuras de hash y particiones, las conexiones se redirigen al nodo derecho para las operaciones de lectura y escritura de datos.

Por lo tanto, las claves no se distribuyen uniformemente a cada partición de una instancia. Si una partición contiene una clave grande o una clave de mucho uso, la capacidad o el rendimiento de la partición se sobrecargará, pero la carga en otras particiones sigue siendo baja. Como resultado, no se alcanza el cuello de botella de la capacidad o del rendimiento de toda la instancia.

3.32 ¿DCS admite los complementos, extensiones o módulos externos?

No. DCS for Redis no soporta los complementos, extensiones o módulos externos. No hay plan para apoyar módulos.

3.33 ¿Por qué desaparece una clave en Redis?

Normalmente, las claves de Redis no desaparecen. Si falta una clave, es posible que haya caducado, que haya sido desalojada o que haya sido eliminada.

Realice las siguientes comprobaciones una por una:

1. Compruebe si la clave ha caducado.
2. Vea la información de monitoreo y verifique si se ha activado el desalojo.
3. Ejecute el comando **INFO** en el lado del servidor para comprobar si se ha eliminado la clave.

3.34 ¿Por qué ocurre un error de OOM durante una conexión de Redis?

Síntomas

"Error in execution; nested exception is io.lettuce.core.RedisCommandExecutionException: OOM command not allowed when used memory > 'maxmemory'" se devuelve durante una conexión de Redis.

Localización de fallas

Un error fuera de memoria (OOM) indica que se ha superado el máximo de memoria. En la información de error, el parámetro **maxmemory** indica la memoria máxima configurada en el servidor de Redis.

Si el uso de memoria de la instancia de Redis es inferior al 100%, la memoria del nodo en el que se escriben los datos puede haber alcanzado el límite máximo. Conéctese a cada nodo del clúster mediante la ejecución de **redis-cli -h <redis_ip> -p 6379 -a <redis_password> -c --bigkeys**. Cuando se conecte a un nodo de réplica, ejecute el comando **READONLY** antes de ejecutar el comando **bigkeys**.

3.35 ¿Qué clientes puedo utilizar para Clúster Redis en diferentes lenguajes de programación?

La siguiente tabla compara Clúster Redis y Clúster Proxy en DCS.

Tabla 3-4 Comparación de Clúster Redis y Clúster Proxy

Concepto	Clúster Redis	Clúster Proxy
Compatibilidad con Redis	Alto	Medio
Compatibilidad del cliente	Medio (El modo de clúster debe estar habilitado en el cliente.)	Alto
Costos	Alto	Medio
Latencia	Bajo	Medio
Separación de lectura/escritura	Soporte nativo (configuración del SDK del cliente)	Desplegado mediante el uso de proxy
Rendimiento	Alto	Medio

Clúster Redis no utiliza proxy y, por lo tanto, ofrece una menor latencia y un mayor rendimiento. Sin embargo, las instancias de Clúster Redis se basan en el protocolo de código abierto de Clúster Redis, por lo que su compatibilidad con el cliente es menor que la de las instancias de Clúster Proxy.

En la siguiente tabla se enumeran los clientes que se pueden utilizar para Clúster Redis.

Tabla 3-5 Clientes que pueden ser utilizados para Clúster Redis

Idioma	Cliente	Documentos de referencia
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**

Idioma	Cliente	Documentos de referencia
Go	Go Redis	Clúster Redis: **https://pkg.go.dev/github.com/go-redis/redis/v8#NewClusterClient** Proxy Clúster, nodo único o principal/en 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**

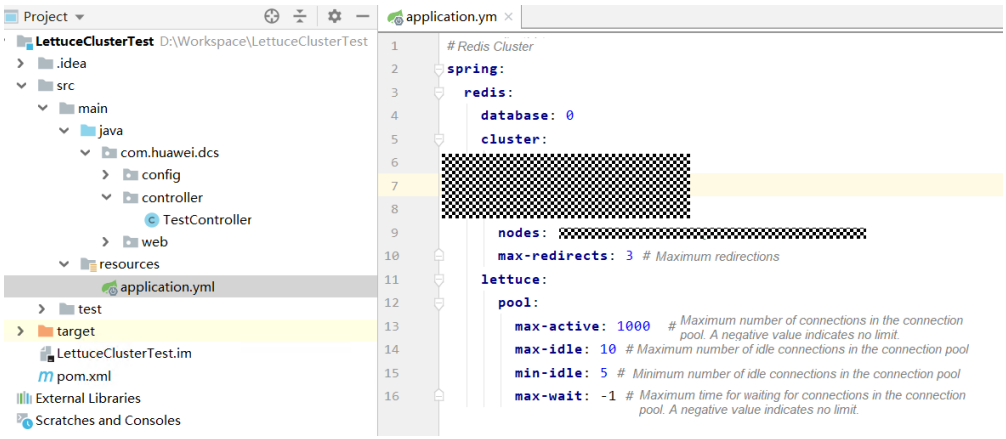
Para ver todos los clientes de Redis, consulte **https://redis.io/clients**.

3.36 ¿Por qué necesito configurar el tiempo de espera para Clúster Redis?

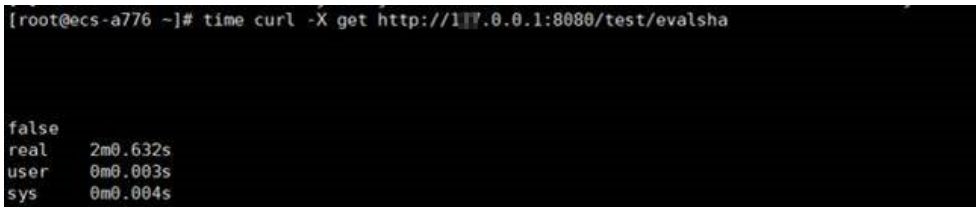
Si el tiempo de espera no está configurado, las conexiones fallarán.

Cuando se conecta a una instancia de Clúster Redis mediante Spring Boot y Lettuce, conéctese a todos los nodos del clúster, incluidas las réplicas defectuosas.

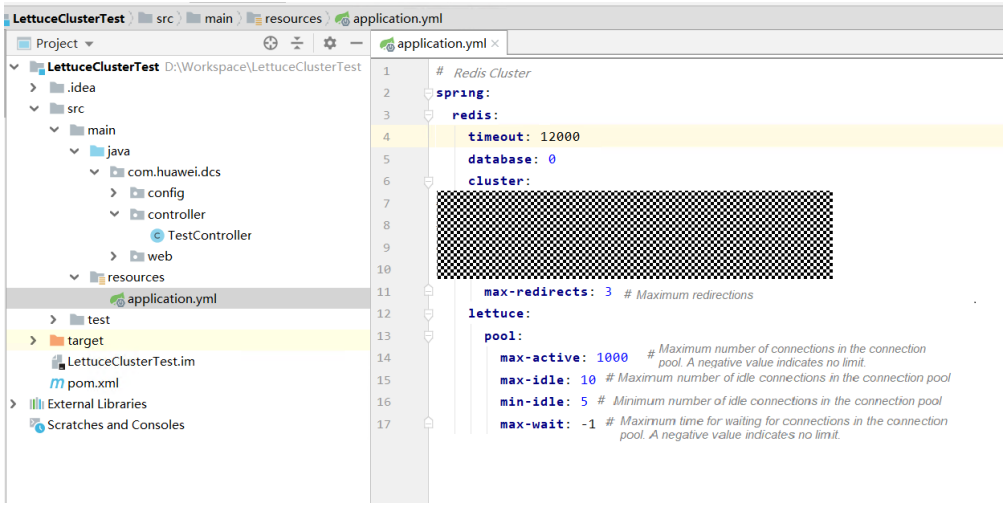
- Si el tiempo de espera no está configurado, el bloqueo a nivel de minutos (120s en versiones anteriores de Lettuce y 60s en la nueva versión) puede ocurrir cuando hay una réplica defectuosa, como se muestra en la siguiente figura.



El tiempo de acceso al servicio de extremo a extremo puede alcanzar el tiempo de espera máximo, como se muestra en la siguiente figura.



- Después de configurar el parámetro **timeout** en el cliente, el tiempo de espera en la réplica se acortará en gran medida. Puede ajustar el tiempo de espera en función de los requisitos de servicio. Se supone que la configuración es la siguiente.



La siguiente figura muestra el tiempo de acceso al servicio de extremo a extremo después de completar la configuración.

Si el parámetro **timeout** no está configurado y hay un nodo defectuoso, las conexiones del cliente se bloquearán.

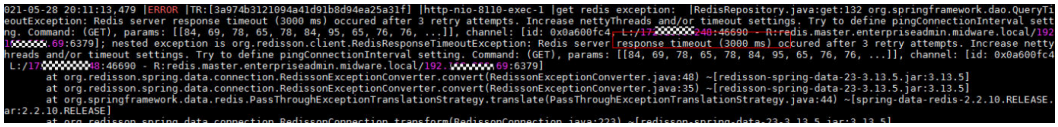
Configure el tiempo de espera según lo que el servicio puede tolerar. Por ejemplo, si necesita solicitar Redis dos veces en una solicitud de HTTP y el tiempo de espera máximo de una

solicitud de HTTP es 10s, se recomienda que establezca el tiempo de espera en Redis en 5s. Esto evita la interrupción del servicio si se producen fallas debido a una larga duración de tiempo de espera o a la ausencia de tiempo de espera.

3.37 ¿Por qué veo un error de tiempo de espera al leer datos de Redis?

Síntomas

Cuando lee datos de Redis, se devuelve el error de tiempo de espera "redis server response timeout (3000 ms) occurred after 3 retry attempts".



Solución de problemas

1. Aumente el tiempo de espera basado en la información de error.
2. Compruebe si el error se devuelve después de realizar una operación en una clave grande. Se recomienda que el tamaño de la clave no sea superior a 10 KB.

   Redis limita el tamaño de cada valor de cadena a 512 MB. En el desarrollo real, mantenga el tamaño dentro de 10 KB. De lo contrario, CPU y NIC estarán muy cargadas.

   Mantenga el número de Hashes, Lists, Sets o Zsets dentro de 5000.

   Teóricamente, el número de elementos en cada HashSet debe ser inferior a 2^32.
3. Aumente el valor del parámetro **PingConnectionInterval** en función de la información de error.

3.38 ¿Cuáles son las limitaciones en el despliegue de multi-BD en una instancia de Clúster Proxy?

Tenga en cuenta las siguientes restricciones cuando considere desplegar multi-BD:

- **Restricciones de uso:**
  a. El comando **SWAPDB** no admite multi-BD.
  b. El comando **INFO KEYSPACE** no devuelve datos de multi-BD.
  c. Para consultar el número total de claves de cada base de datos, utilice el comando personalizado **dbstats**. El uso de la CPU aumentará en el nodo que ejecuta este comando.
  d. Los scripts de LUA no soportan multi-BD.
  e. El comando **RANDOMKEY** no admite multi-BD.
  f. El comando **SELECT** no se puede incrustar en transacciones.
  g. **PUBLISH** no se puede usar en scripts de Lua.
  h. El número de base de datos varía de 0 a 255.

i. Las instancias de Clúster Proxy de DCS Redis 3.0 no soportan multi-BD.

- **Restricciones de rendimiento**

  a. El comando **FLUSHDB** elimina las claves una por una, lo que lleva mucho tiempo y es más lento que el despliegue nativo del código abierto. La velocidad de ejecución del comando **FLUSHDB** es la misma que la del comando **SCAN** (que debe ser probado por el cliente).

  b. El comando **DBSIZE** consume mucho tiempo. No lo utilice en el código.

  c. Si se utiliza multi-BD, el rendimiento de los comandos **KEYS** y **SCAN** se deteriora hasta en un 50%.

- **Otras restricciones**:

  El almacenamiento de backend reescribe las claves basadas en ciertas reglas. Las claves del archivo de RDB exportado no son las claves originales. Sin embargo, el acceso a través del protocolo de Redis no se ve afectado.

Procedimiento para habilitar multi-BD en una instancia de una sola BD

De forma predeterminada, multi-BD está deshabilitada. Antes de habilitar o deshabilitar multi-BD para una instancia, borre los datos de la instancia. Haga lo siguiente para habilitar multi-BD.

Paso 1 Inicie sesión en la consola de DCS.

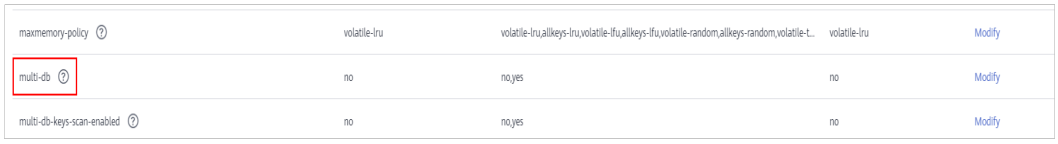
Paso 2 Conéctese a la instancia y ejecute el comando **FLUSHALL** para borrar los datos de la instancia.

Paso 3 En la página **Cache Manager** de la consola de DCS, haga clic en la instancia de DCS deseada.

Paso 4 Elija **Instance Configuration** > **Parameters**.

Paso 5 Haga clic en **Modify** en la fila que contiene el parámetro **multi-db** y, a continuación, cambie su valor a **yes**.

Paso 6 Haga clic en **Save** y, a continuación, confirme la modificación. No es necesario reiniciar la instancia.



----Fin

3.39 ¿Puedo cambiar la AZ de una instancia?

No.

Si desea utilizar una AZ diferente, cree otra instancia en la AZ deseada y, a continuación, migre los datos.

NOTA

- La conmutación IP solo es compatible con las instancias de DCS Redis 4.0 y 5.0.
- La conmutación IP solo se admite cuando las instancias de origen y destino son instancias de Redis en la nube.

Requisitos previos

- La instancia de destino está disponible. Si ya tiene una instancia de DCS Redis, utilícela directamente y borre los datos de instancia antes de la migración. Para obtener más información, consulte la sección **Borrar datos de instancia de DCS**.

  Si los datos de la instancia de destino no se borran antes de la migración y las instancias de origen y destino contienen la misma clave, la clave de la instancia de destino se sobrescribirá después de la migración.
- El Redis de destino, el Redis de origen y los recursos de tareas de migración están en la misma VPC.

  NOTA

  Si las instancias de Redis de destino y de origen no están en la misma VPC, asegúrese de que los recursos de la máquina virtual de la tarea de migración puedan acceder a estas instancias.

  - Si las instancias de Redis de origen y destino están en la misma región, cree una interconexión de VPC haciendo referencia a **Interconexión de VPC**.
  - Si las instancias de origen y destino de Redis se encuentran en diferentes regiones, cree una conexión a la nube consultando **Pasos iniciales de Cloud Connect**.
- Las instancias de destino y origen utilizan el mismo puerto.
- La conmutación IP solo se puede realizar cuando se cumplen las siguientes condiciones:
  - La conmutación IP depende de la función de migración de datos. Por lo tanto, las instancias de origen y destino deben admitir la función de migración de datos.
  - En la siguiente tabla se enumeran los escenarios de conmutación IP admitidos.

Tabla 3-6 Escenarios de conmutación de IP

Fuente	Objetivo
Nodo único, separación de lectura/escritura, o principal/en espera	Nodo único, separación de lectura/escritura, principal/en espera o Clúster Proxy
Clúster Proxy	Nodo único, separación de lectura/escritura, principal/en espera o Clúster Proxy

Precauciones para la conmutación de IP

1. La migración en línea se detendrá durante la conmutación.
2. Las instancias serán de solo lectura durante un minuto y se desconectarán durante varios segundos durante la conmutación.
3. Si la aplicación no puede volver a conectarse a Redis o manejar excepciones, es posible que tenga que reiniciar la aplicación después de la conmutación de IP.

4. Si las instancias de origen y destino se encuentran en diferentes subredes, la información de la subred se actualizará después de la conmutación.
5. Si la fuente es una instancia principal/en standby, la dirección IP del nodo en standby no se conmutará. Asegúrese de que sus aplicaciones no utilicen esta dirección IP.
6. Si sus aplicaciones usan un nombre de dominio para conectarse a Redis, el nombre de dominio se usará para la instancia de origen. Seleccione **Yes** para **Switch Domain Name**.
7. Asegúrese de que las contraseñas de las instancias de origen y de destino sean las mismas. Si son diferentes, la verificación fallará después de la conmutación.
8. Si se configura una lista blanca para la instancia de origen, asegúrese de que la misma lista blanca está configurada para la instancia de destino antes de cambiar las direcciones IP.

Conmutación de direcciones IP

Paso 1 Inicie sesión en la **consola DCS**..

Paso 2 Haga clic en en la esquina superior izquierda de la consola de gestión y seleccione la región donde se encuentra la instancia.

Paso 3 En el panel de navegación, elija **Data Migration**.

Paso 4 Haga clic en **Create Online Migration Task**.

Paso 5 Introduzca el nombre y la descripción de la tarea.

Paso 6 Configure la VPC, la subred y el grupo de seguridad para la tarea de migración.

La VPC, la subred y el grupo de seguridad facilitan la migración. Asegúrese de que los recursos de migración puedan acceder a las instancias de Redis de origen y destino.

Paso 7 Configure la tarea de migración haciendo referencia a **Configuración de la tarea de migración en línea**. Establezca **Migration Type** en **Full + Incremental**.

Paso 8 En la página **Online Migration**, cuando el estado de la tarea de migración cambie a **Incremental migration in progress**, elija **More** > **Switch IP** en la columna **Operation**.

Paso 9 En el cuadro de diálogo **Switch IP**, seleccione si desea cambiar el nombre de dominio.

NOTA

- Si se utiliza un nombre de dominio, cámbielo o debe modificar el nombre de dominio en el cliente.
- Si no se utiliza ningún nombre de dominio, se actualizará el DNS de las instancias.

Paso 10 Haz clic en **OK**. La tarea de conmutación de direcciones IP se envía correctamente. Cuando el estado de la tarea de migración cambia a **IP switched**, se completa el cambio de dirección IP.

----Fin

Retroceder las direcciones IP

Si desea cambiar la dirección IP de la instancia a la dirección IP original, realice las siguientes operaciones:

Paso 1 Inicie sesión en la **consola DCS**..

Paso 2 Haga clic en en la esquina superior izquierda de la consola de gestión y seleccione la región donde se encuentra la instancia.

Paso 3 En el panel de navegación, elija **Data Migration**.

Paso 4 En la página **Online Migration**, busque la fila que contiene la tarea de migración en el estado **IP switched**, elija **More** > **Roll Back IP**.

Paso 5 En la ventana de confirmación, haga clic en **Yes**. La tarea de reversión de direcciones IP se envía correctamente. Cuando el estado de la tarea cambia a **IP rolled back**, se completa la reversión.

----Fin

3.40 Explicación y uso de etiquetas de hash

Diseño de etiquetas de hash

Las operaciones de múltiples claves, como las que utilizan el comando **MSET** o los scripts de Lua, son atómicas. Todas las claves especificadas se ejecutan al mismo tiempo. Sin embargo, en un clúster, cada clave es hash a una partición dada, y las operaciones de múltiples claves ya no son atómicas. Las claves pueden asignarse a diferentes ranuras. Como resultado, algunas claves se actualizan, mientras que otras no. Si hay una etiqueta de hash, el clúster determina qué ranura asignar una clave basándose en la etiqueta de hash. Las claves con la misma etiqueta de hash se asignan a la misma ranura.

Uso de etiquetas de hash

Solo el contenido entre la primera llave de apertura ({) y la siguiente primera llave de cierre (}) es hash.

Por ejemplo:

- En las claves **{user1000}.following** y **{user1000}.followers** solo hay un par de llaves. **user1000** se va a hash.
- En la clave **foo{}{bar}**, no hay contenido entre la primera { y la primera }. Toda la clave **foo{}{bar}** será hash como de costumbre.
- En la clave **foo{{bar}}zap**, **{bar** (el contenido entre la primera { y la primera }) es hash.
- En la clave **foo{bar}{zap}**, **bar** es hash porque está entre el primer par de { y }.

Ejemplo de etiqueta de hash

Cuando se realiza la siguiente operación:

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

Se muestra el siguiente error:

ERR 'key1' y 'key2' no están en la misma ranura

Puede usar una etiqueta de hash para resolver este problema:

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


3.41 ¿Se conservarán los datos almacenados en caché después de reiniciar una instancia?

Después de reiniciar una instancia de DCS del nodo único, se eliminan los datos de la instancia.

Las instancias principal/en espera y de clúster (excepto los clústeres de réplica única) admiten la persistencia de AOF de forma predeterminada. Los datos se conservan después de reiniciar estas instancias.

Si la persistencia de AOF está desactivada (**appendonly** está establecido en **no**), los datos se eliminan después de reiniciar las instancias.

3.42 ¿Cómo puedo comprar una instancia de multi-BD de Clúster Proxy?

Cuando usted compra una instancia de Clúster Proxy, solo hay una base de datos por defecto. Esta sección describe cómo comprar una instancia de Clúster Proxy con múltiples bases de datos.

NOTA

Antes de comenzar, aprenda sobre **las limitaciones en el despliegue de multi-BD**.

Paso 1 Inicie sesión en la **consola de DCS**.

Paso 2 Haga clic en en la esquina superior izquierda para seleccionar una región.

Paso 3 En el panel de navegación, elija **Parameter Templates**.

Paso 4 En la fila que contiene la plantilla con la versión del motor de caché y el tipo de instancia deseados (Clúster Proxy), haga clic en **Customize**.

Paso 5 Ajusta **multi-db** a **yes**.

Paso 6 Ingrese un nuevo nombre de plantilla y haga clic en **OK**. La plantilla personalizada se crea correctamente.

Paso 7 En el panel de navegación, elija **Cache Manager**. A continuación, haga clic en **Buy DCS Instance** para crear una instancia de Clúster Proxy.

Establezca **Parámetro Configuration** en **Use custom template** y seleccione la plantilla personalizada creada en el paso anterior.

Una vez creada la instancia, conéctese a ella para comprobar si tiene varias bases de datos.

----Fin

3.43 ¿Por qué se congela una instancia?

Las instancias están en el estado **Frozen** si el paquete anual/mensual no se renueva después de la expiración. Las instancias congeladas siguen en ejecución, pero no se pueden utilizar hasta que se renueve el paquete.

4 Escalamiento y actualización de instancias

4.1 ¿Puedo actualizar la versión para una instancia de DCS Redis, por ejemplo, de Redis 4.0 a Redis 5.0?

No. Las diferentes versiones de Redis utilizan diferentes arquitecturas subyacentes. La versión de Redis utilizada por una instancia de DCS no se puede cambiar una vez creada la instancia.

Si el servicio requiere las características de versiones superiores de Redis, cree una nueva instancia de DCS Redis de una versión superior y, a continuación, migre los datos de la instancia original a la nueva. Para obtener más información sobre cómo migrar datos, consulte la **Guía de migración de datos**.

4.2 ¿Se interrumpen los servicios si se realiza el mantenimiento durante la ventana de tiempo de mantenimiento?

El personal de O&M se pondrá en contacto con usted antes de realizar el mantenimiento durante la ventana de tiempo de mantenimiento, informándole de las operaciones y sus impactos. No es necesario preocuparse por las excepciones de ejecución de instancia.

4.3 ¿Las instancias se detienen o se reinician durante la modificación de la especificación?

No. Las modificaciones de especificación pueden tener lugar mientras la instancia se está ejecutando y no afectan a ningún otro recurso.

4.4 ¿Qué cambios de tipo de instancia de DCS son compatibles?

Tabla 4-1 Opciones de cambio de tipo de instancia admitidas por diferentes instancias de DCS

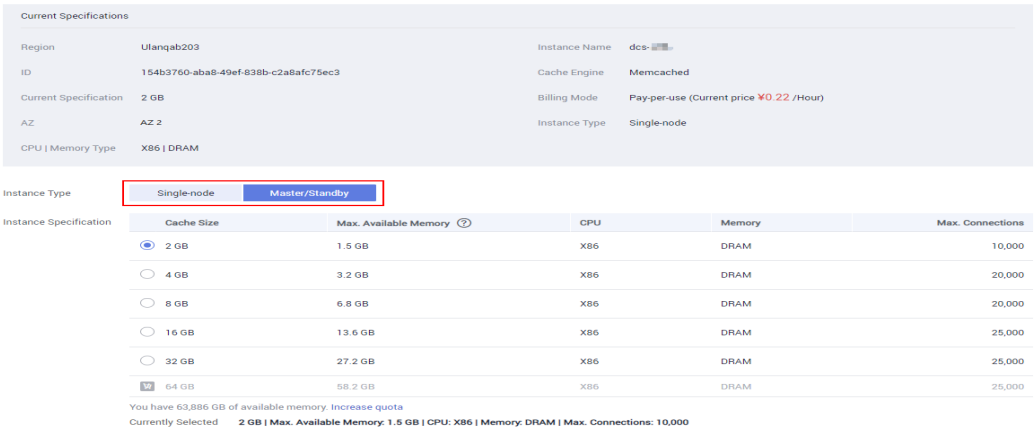
Versión	Cambio de tipo admitido	Precauciones
Redis 3.0	De nodo único a principal/en standby	La instancia no se puede conectar durante varios segundos y permanece de sólo lectura durante aproximadamente un minuto.
	De principal/en standby a Clúster Proxy	1. Si los datos de una instancia de DCS Redis 3.0 principal/en standby se almacenan en múltiples bases de datos, o en bases de datos que no son DB0, la instancia no se puede cambiar al tipo de proxy Clúster. Una instancia principal/en standby se puede cambiar al tipo Clúster Proxy solo si sus datos se almacenan solo en DB0. 2. La instancia no se puede conectar y permanece de sólo lectura durante 5 a 30 minutos.
Memcached	De nodo único a principal/en standby	Los servicios se interrumpen durante varios segundos y permanecen de solo lectura durante aproximadamente 1 minuto.

Versión	Cambio de tipo admitido	Precauciones
Redis 4.0/5.0	De principal/en standby a Clúster Proxy De la separación de lectura/escritura a Clúster Proxy De Clúster Proxy a principal/en standby De Clúster Proxy a la separación de lectura/escritura	1. Antes de cambiar el tipo de instancia a Clúster Proxy, evalúe el impacto en los servicios. Para obtener más información, consulte **¿Cuáles son las restricciones en la implementación de Multi-DB en una instancia Clúster Proxy?** y **Restricciones de comando**. 2. El uso de memoria debe ser inferior al 70% de la memoria máxima de la nueva variante. 3. Algunas claves pueden ser desalojadas si el uso actual de la memoria excede el 90% del total. 4. Después del cambio, **cree reglas de alarma** de nuevo para la instancia. 5. Para las instancias que actualmente son principales/en standby, asegúrese de que su dirección IP o nombre de dominio de solo lectura no sean utilizados por su aplicación. 6. Si la aplicación no puede volver a conectarse a Redis o controlar las excepciones, es posible que tenga que reiniciar la aplicación después del cambio. 7. Modifique las especificaciones de las instancias en períodos de poca actividad. Una instancia se interrumpe temporalmente y permanece en estado de solo lectura por aproximadamente 1 minuto durante el cambio de especificaciones.

Para obtener más información sobre los comandos admitidos por diferentes tipos de instancias, consulte **Compatibilidad de comandos**.

No se admiten los cambios de tipo de instancia que no se enumeran en la tabla anterior. Para modificar las especificaciones mientras se cambia el tipo de instancia, consulte **Conmutación de IP**.

Para comprobar si puede cambiar el tipo de instancia de una instancia, consulte los parámetros que se muestran en la página **Modify Specifications** de la consola de DCS. El siguiente escenario muestra que se puede cambiar el tipo de instancia.



4.5 ¿Se interrumpen los servicios durante la modificación de la especificación?

Modifique las especificaciones de las instancias en períodos de poca actividad.

Si la modificación falló en las horas pico (por ejemplo, cuando el uso de memoria o CPU es superior al 90% o cuando el tráfico de escritura aumenta), inténtalo de nuevo durante las horas no pico.

En la siguiente tabla se describe el impacto de la modificación de la especificación.

Cambio del tipo de instancia

Tabla 4-2 Opciones de cambio de tipo de instancia admitidas por diferentes instancias de DCS

Versión	Cambio de tipo admitido	Precauciones
Redis 3.0	De nodo único a principal/en standby	La instancia no se puede conectar durante varios segundos y permanece de sólo lectura durante aproximadamente un minuto.
	De principal/en standby a Clúster Proxy	1. Si los datos de una instancia de DCS Redis 3.0 principal/en standby se almacenan en múltiples bases de datos, o en bases de datos que no son DB0, la instancia no se puede cambiar al tipo de proxy Clúster. Una instancia principal/en standby se puede cambiar al tipo Clúster Proxy solo si sus datos se almacenan solo en DB0. 2. La instancia no se puede conectar y permanece de sólo lectura durante 5 a 30 minutos.
Memcached	De nodo único a principal/en standby	Los servicios se interrumpen durante varios segundos y permanecen de solo lectura durante aproximadamente 1 minuto.

Versión	Cambio de tipo admitido	Precauciones
Redis 4.0/5.0	De principal/en standby a Clúster Proxy De la separación de lectura/escritura a Clúster Proxy De Clúster Proxy a principal/en standby De Clúster Proxy a la separación de lectura/escritura	1. Antes de cambiar el tipo de instancia a Clúster Proxy, evalúe el impacto en los servicios. Para obtener más información, consulte **¿Cuáles son las restricciones en la implementación de Multi-DB en una instancia Clúster Proxy?** y **Restricciones de comando**. 2. El uso de memoria debe ser inferior al 70% de la memoria máxima de la nueva variante. 3. Algunas claves pueden ser desalojadas si el uso actual de la memoria excede el 90% del total. 4. Después del cambio, **cree reglas de alarma** de nuevo para la instancia. 5. Para las instancias que actualmente son principales/en standby, asegúrese de que su dirección IP o nombre de dominio de solo lectura no sean utilizados por su aplicación. 6. Si la aplicación no puede volver a conectarse a Redis o controlar las excepciones, es posible que tenga que reiniciar la aplicación después del cambio. 7. Modifique las especificaciones de las instancias en períodos de poca actividad. Una instancia se interrumpe temporalmente y permanece en estado de solo lectura por aproximadamente 1 minuto durante el cambio de especificaciones.

No se admiten los cambios de tipo de instancia que no se enumeran en la tabla anterior. Para modificar las especificaciones mientras se cambia el tipo de instancia, consulte **Conmutación IP**.

Ajuste de escala

- Opciones de escala

Tabla 4-3 Opciones de escala compatibles con diferentes instancias

Motor de memoria caché	Nodo único	Principal/En standby	Clúster Redis	Clúster Proxy	Separación de lecturas/ escrituras
Redis 3.0	Escalar hacia arriba/ hacia abajo	Escalar hacia arriba/ hacia abajo	Escalar hacia arriba/ hacia abajo	Escalando hacia arriba	-
Redis 4.0	Escalar hacia arriba/ hacia abajo	Escalado hacia arriba/ abajo, hacia fuera/entra	Escalado hacia arriba/ abajo, hacia fuera/entra	Escalar hacia arriba/hacia abajo	Escalado hacia arriba/ abajo, hacia fuera/entra
Redis 5.0	Escalar hacia arriba/ hacia abajo	Escalado hacia arriba/ abajo, hacia fuera/entra	Escalado hacia arriba/ abajo, hacia fuera/entra	Escalar hacia arriba/hacia abajo	Escalado hacia arriba/ abajo, hacia fuera/entra
Memcached	Escalar hacia arriba/ hacia abajo	Escalar hacia arriba/ hacia abajo	-	-	-
Redis 6.0 edición básica	Escalar hacia arriba/ hacia abajo	Escalar hacia arriba/ hacia abajo	-	-	-
Redis 6.0 ediciones profesionales	-	No se admite ningún cambio.	-	-	-

NOTA

Si la memoria reservada de una instancia de DCS Redis 3.0 o Memcached es insuficiente, la modificación puede fallar cuando se agota la memoria. Para obtener más información, consulte **Memoria reservada**.

- Impacto de la escala

Tabla 4-4 Impacto de la escala

Tipo de instancia	Tipo de escala	Impacto
Nodo único, principal/en standby y separación de lectura/escritura	Escalar hacia arriba/hacia abajo	● Una instancia de DCS Redis 4.0 o 5.0 se desconectará durante varios segundos y permanecerá de sólo lectura durante aproximadamente 1 minuto. Una instancia de DCS Redis 3.0 se desconectará y permanecerá de sólo lectura durante 5 a 30 minutos. ● Para escalar, solo se expande la memoria de la instancia. La capacidad de procesamiento de la CPU no se mejora. ● Las instancias DCS de nodo único no admiten persistencia de datos. Los datos no se conservan durante el escalado. Después de escalar, compruebe si los datos están completos e importe los datos si es necesario. Si hay datos importantes, utilice una herramienta de migración para migrar los datos a otras instancias para realizar copias de seguridad. ● Los registros de copia de seguridad de instancias principal/en standby y de separación lectura/escritura no se pueden restaurar después de escalar.

Tipo de instancia	Tipo de escala	Impacto
Clúster Proxy y Clúster Redis	Escalar hacia arriba/hacia abajo	● El escalado implica la migración de datos, lo que aumenta la latencia de acceso. Para una instancia de Clúster Redis, asegúrese de que el cliente puede procesar correctamente los comandos MOVED y ASK. De lo contrario, las solicitudes fallarán. ● Si la memoria se llena durante el escalado debido a que se escribe una gran cantidad de datos, el escalado fallará. ● No se pueden restaurar los registros de copia de seguridad creados antes de la escala. ● Antes de escalar, compruebe si hay claves grandes a través de Análisis de caché. Redis tiene un límite en la migración de claves. Si la instancia tiene una clave única superior a 512 MB, el escalado fallará cuando se agote el tiempo de migración de clave grande entre nodos. Cuanto más grande sea la clave, más probabilidades hay de que la migración falle. ● Antes de escalar hacia arriba o hacia abajo una instancia de Clúster Redis, asegúrese de que la actualización automatizada de la topología del clúster esté habilitada si usa Lettuce. Si está deshabilitado, tendrá que reiniciar el cliente después de escalar. Para más detalles sobre cómo habilitar la actualización automatizada, vea **un ejemplo de uso de Lettuce para conectarse a una instancia de Clúster Redis**. ● El escalamiento no interrumpe las conexiones, sino que ocupará los recursos de la CPU, lo que reducirá el rendimiento hasta en un 20%. ● Durante la ampliación, se agregan nuevos nodos del servidor Redis y los datos se equilibran automáticamente en los nuevos nodos. ● Para reducir la escala de una instancia, asegúrese de que la memoria utilizada de cada nodo sea inferior al 70% de la memoria máxima por nodo del nuevo sabor. ● Si la cantidad de particiones disminuye durante la reducción de la escala, los nodos se eliminarán. Antes de reducir la escala, asegúrese de que los nodos eliminados no se referencian directamente en la aplicación, para evitar excepciones de acceso al servicio. ● Si la cantidad de particiones disminuye durante la reducción de la escala, los nodos se eliminarán y las conexiones se interrumpirán. Si la aplicación no puede volver a conectarse a Redis o controlar las excepciones, es posible que tenga que reiniciar la aplicación después de escalar.

Tipo de instancia	Tipo de escala	Impacto
Instancias principal/en standby, de separación de lectura/escritura y de Clúster Redis	Escalado de salida/entrada (cambio de la cantidad de réplicas)	• Antes de escalar hacia fuera o entra una instancia de Clúster Redis, asegúrese de que la actualización automatizada de la topología del clúster esté habilitada si usa Lettuce. Si está deshabilitado, tendrá que reiniciar el cliente después de escalar. Para más detalles sobre cómo habilitar la actualización automatizada, vea **un ejemplo de uso de Lettuce para conectarse a una instancia de Clúster Redis**. • La eliminación de réplicas interrumpe las conexiones. Si la aplicación no puede volver a conectarse a Redis o manejar excepciones, debe reiniciar la aplicación después de escalar. • Si el número de réplicas ya es el mínimo admitido por la instancia, ya no podrá eliminar réplicas.

4.6 ¿Por qué no puedo modificar las especificaciones de una instancia de DCS?

- Compruebe si se están ejecutando otras tareas.

  Las especificaciones de una instancia de DCS no se pueden modificar si se está ejecutando otra tarea de la instancia. Por ejemplo, no puede eliminar ni escalar una instancia mientras se está reiniciando. Del mismo modo, no puede eliminar una instancia mientras se está ampliando.

  Si la modificación de la especificación falla, inténtelo de nuevo más tarde. Si vuelve a fallar, póngase en contacto con el soporte técnico.
- Cuando cambie una instancia principal/en espera al tipo de Clúster Proxy, compruebe si los datos existen en bases de datos distintas de DB0. La modificación de la especificación fallará si una base de datos distinta de DB0 contiene datos.

  Una instancia principal/en espara se puede cambiar al tipo de Clúster Proxy cuando los datos solo existen en DB0.

4.7 ¿Cómo puedo reducir la capacidad de una instancia de DCS?

Tabla 4-5 enumera las opciones de ajuste compatibles con diferentes instancias de DCS.

Tabla 4-5 Opciones de escala compatibles con diferentes instancias

Motor de memoria caché	Nodo único	Principal/En standby	Clúster Redis	Clúster Proxy	Separación de lecturas/escrituras
Redis 3.0	Escalar hacia arriba/hacia abajo	Escalar hacia arriba/hacia abajo	Escalar hacia arriba/hacia abajo	Escalando hacia arriba	-
Redis 4.0	Escalar hacia arriba/hacia abajo	Escalado hacia arriba/abajo, hacia fuera/entra	Escalado hacia arriba/abajo, hacia fuera/entra	Escalar hacia arriba/hacia abajo	Escalado hacia arriba/abajo, hacia fuera/entra
Redis 5.0	Escalar hacia arriba/hacia abajo	Escalado hacia arriba/abajo, hacia fuera/entra	Escalado hacia arriba/abajo, hacia fuera/entra	Escalar hacia arriba/hacia abajo	Escalado hacia arriba/abajo, hacia fuera/entra
Memcached	Escalar hacia arriba/hacia abajo	Escalar hacia arriba/hacia abajo	-	-	-
Redis 6.0 edición básica	Escalar hacia arriba/hacia abajo	Escalar hacia arriba/hacia abajo	-	-	-
Redis 6.0 ediciones profesionales	-	No se admite ningún cambio.	-	-	-

Para obtener más información sobre cómo cambiar la capacidad, consulte la sección **Modificación de las especificaciones**.

Si desea utilizar una instancia de Clúster Proxy de DCS Redis 3.0 más pequeña, haga una copia de respaldo de los datos de la instancia existente y cree una nueva instancia de Clúster Proxy con la capacidad deseada. A continuación, importe los datos de copia de respaldo a la nueva instancia. Una vez completada la migración de datos, elimine la instancia antigua. Para obtener más información sobre las operaciones de migración de datos, consulte la sección **Importación de archivos de copia de respaldo**.

4.8 ¿Cómo agrego particiones a una instancia de DCS Redis de clúster sin cambiar la memoria?

Después de crear una instancia de Clúster Proxy o de Clúster Redis, puede reducir la capacidad de cada partición y agregar más particiones sin cambiar la memoria total.

Por ejemplo, si una instancia de 8 GB tiene 4 particiones y cada partición tiene 2 GB, puede reducir el tamaño de partición a 1 GB y aumentar la cantidad de partición a 8.

NOTA

No se puede cambiar el tamaño de partición de 1 GB.

Procedimiento

Paso 1 Inicie sesión en la **consola de DCS**.

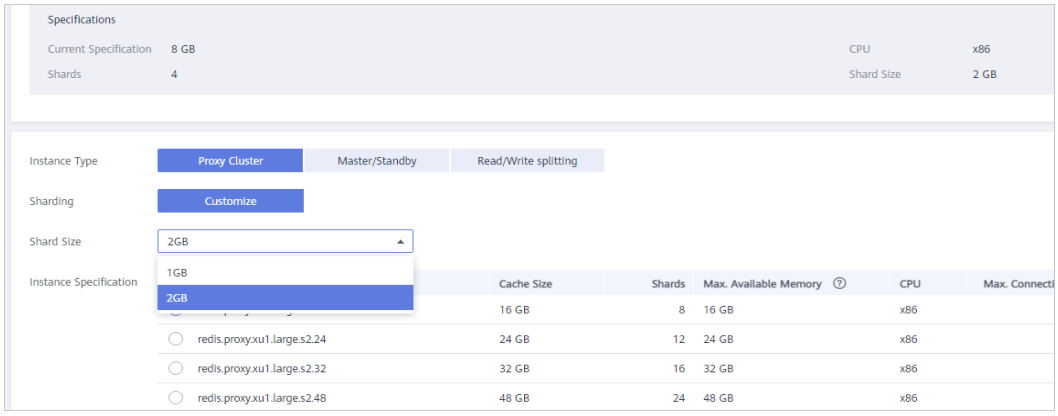
Paso 2 Haga clic en en la esquina superior izquierda para seleccionar una región y un proyecto.

Paso 3 En el panel de navegación, elija **Cache Manager**.

Paso 4 Seleccione **More** > **Modify Specifications** en la fila que contiene la instancia de DCS deseada.

Paso 5 En la página **Modify Instance Specifications** mostrada, especifique **Shard Size** y **Instance Specification**.

Figura 4-1 Selección de un tamaño de partición



Paso 6 Haga clic en **Next**, confirme los detalles y haga clic en **Submit**.

La modificación tarda aproximadamente de 5 a 30 minutos en completarse. Una vez que la modificación se realiza correctamente, el estado de la instancia cambia a **Running**.

----Fin

4.9 ¿Cómo puedo manejar un error cuando uso Lettuce para conectarme a una instancia de Clúster Redis después de la modificación de la especificación?

Síntomas

Si la cantidad de partición cambia durante la modificación de la especificación de una instancia de Clúster Redis, algunas ranuras se migran a nuevas particiones. El siguiente error ocurre cuando usa Lettuce para conectarse a la instancia.

Figura 4-2 Error

```
org.springframework.data.redis.RedisSystemException: Redis exception; nested exception is io.lettuce.core.RedisException: io.lettuce.core.R
edisException: java.lang.IllegalArgumentException: Connection to 192.168.70.125:6379 not allowed. This connection point is not known in the
 cluster view
```

Para obtener más información, consulte **No se permite la conexión a X. Este punto de conexión no se conoce en la vista de clúster**.

Análisis

Proceso de modificación de especificación de una instancia de Clúster Redis:

Después de iniciarse, el cliente obtiene la topología del nodo de clúster mediante el comando **Cluster Nodes** basado en RESP2, y mantiene la topología en su estructura de datos en memoria.

Para el acceso a datos, el cliente utiliza el algoritmo de CRC16 para calcular el intervalo hash de una clave, y automáticamente encamina las solicitudes basadas en la topología y la información de intervalo almacenada en la memoria.

Si el número de particiones cambia durante la escala, la topología y la asignación de ranuras cambian. En este caso, el cliente necesita actualizar automáticamente la topología. De lo contrario, la ruta de solicitud puede fallar o la ubicación de la ruta puede ser incorrecta. Como resultado, se notifica un error durante la conexión del cliente.

Por ejemplo, cuando el número de fragmentos en una instancia de Clúster Redis cambia de tres a seis, la topología y el mapeo de ranuras cambian como se muestra en las siguientes figuras.

Figura 4-3 Una instancia de Clúster Redis antes de escalar

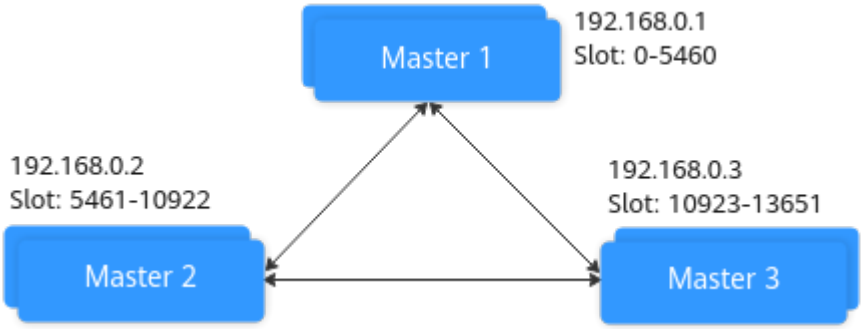
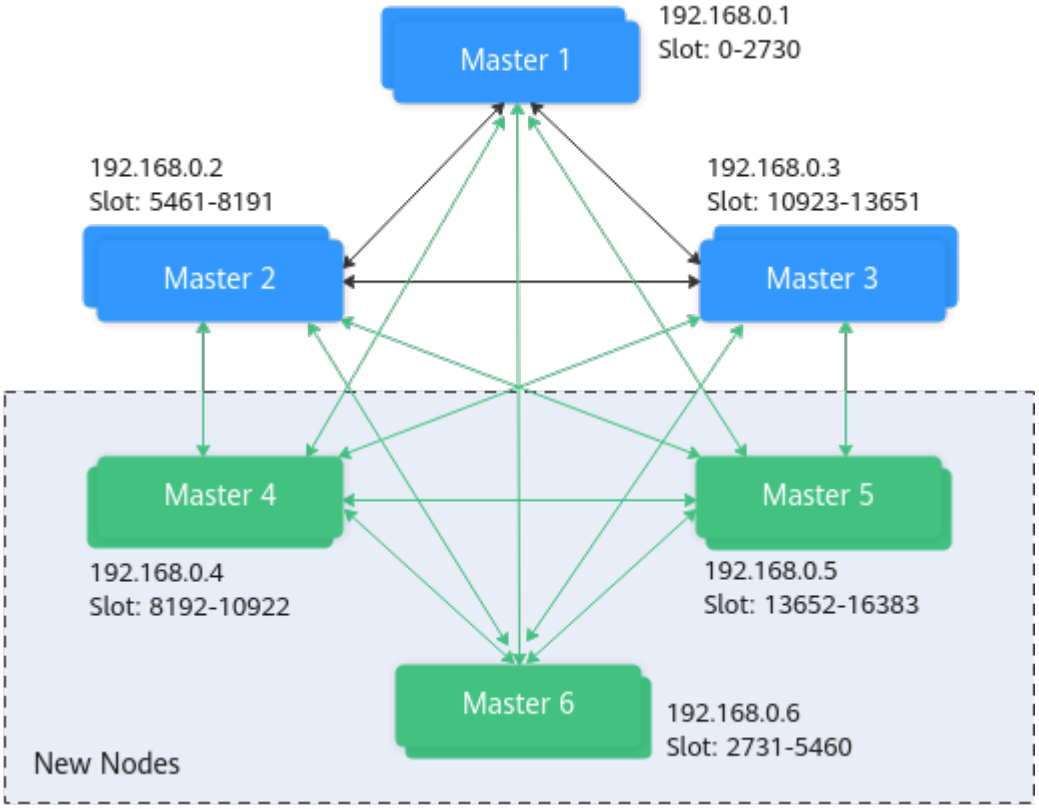


Figura 4-4 Una instancia de Clúster Redis después de escalar



Soluciones

Solución 1 (Recomendada)

Habilite la actualización automatizada de la topología.

```
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 más detalles, vea **un ejemplo de uso de Lettuce para conectarse a una instancia de Clúster Redis**.

NOTA

Si utiliza Lettuce para conectarse a una instancia de Clúster Redis y la actualización automatizada no está habilitada, debe reiniciar el cliente después de la modificación de la especificación.

Solución 2

Deshabilite la validación de la pertenencia al nodo del clúster.

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


Si **validateClusterNodeMembership** es **true**, compruebe si la dirección de conexión actual está en la topología de clúster obtenida a través de **CLUSTER NODES**, antes de conectarse al clúster. Si no está en la topología, se produce el error.

NOTA

Impacto de la desactivación de la validación de la pertenencia al nodo del clúster:

- Falta de detección de brechas de seguridad.
- Si se desactiva la actualización de topología automatizada, se puede generar una solicitud de redireccionamiento MOVED después de que se cambien las especificaciones de Clúster Redis y se aumente la cantidad de partición. La redirección aumenta la carga de red del clúster y el tiempo necesario para procesar una sola solicitud. Si la cantidad de partición disminuye, las particiones eliminadas no se pueden conectar.

4.10 ¿Puedo expandir una partición única de una instancia de clúster?

No. Solo puede agregar más partición para ampliar la capacidad de la instancia.

Si desea utilizar un tamaño más grande en cada partición de una instancia de Clúster Proxy, cambie la instancia al tipo principal/en espera, y luego cambie de nuevo a Clúster Proxy con el tamaño de partición deseado. Antes de cambiar el tipo de instancia a Clúster Proxy, evalúe el impacto en los servicios. Para más detalles, véase **¿Cuáles son las limitaciones en el despliegue de multi-BD en una instancia de Clúster Proxy?**

5 Copia de seguridad, exportación y migración de datos

5.1 ¿Cómo puedo exportar datos de instancia de DCS Redis?

- Instancias principal/en espera, de separación de lectura/escritura y de clúster:

  Estas instancias admiten las copias de respaldo. Realice las siguientes operaciones para exportar datos:

  a. En la página **Backups and Restorations**, vea las tareas de copia de respaldo.
  b. Si no hay copia de respaldo, cree una copia de respaldo y descargue el archivo de copia de respaldo como se le indique.

  NOTA

  Si sus instancias de DCS se crearon hace mucho tiempo, es posible que las versiones de estas instancias no sean lo suficientemente avanzadas como para admitir algunas funciones nuevas (como backup y restauración). Puede ponerse en contacto con el soporte técnico para actualizar sus instancias de DCS. Después de la actualización, puede realizar copias de respaldo y restaurar las instancias.

- Instancias de nodo único:

  Las instancias de nodo único no admiten la función de copia de respaldo. Puede utilizar redis-cli para exportar datos a archivos de RDB. Esta operación depende del comando **SYNC**.

  – Si la instancia permite el comando **SYNC** (como una instancia de un solo nodo de Redis 3.0), ejecute el siguiente comando para exportar los datos de la instancia:

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

  – Si la instancia no permite el comando **SYNC** (como una instancia de nodo único de Redis 4.0 o 5.0), migre los datos de la instancia a una instancia principal/en espera y exporte los datos mediante la función de copia de respaldo.

5.2 ¿Por qué no se modifica la memoria de una instancia de DCS Redis después de la migración de datos mediante Rump, incluso si no se devuelve ningún mensaje de error?

Para obtener más información sobre cómo usar Rump, consulte la **Guía de migración de datos**.

Causas posibles:

- Rump no admite la migración a instancias de clúster de DCS.
- Los comandos se ejecutan incorrectamente en Rump.

5.3 ¿Puedo exportar datos de copia de respaldo de instancias de DCS Redis a los archivos de RDB en la consola?

- Instancias de DCS Redis 3.0

  No. En la consola, los datos de copia de respaldo de una instancia de DCS Redis 3.0 solo se pueden exportar a los archivos de AOF. Para exportar datos a los archivos de RDB, ejecute el siguiente comando en redis-cli:

  redis-cli -h *{redis_address}* -p 6379 [-a *password*] --rdb {output.rdb}
- Instancias de DCS Redis 4.0 y 5.0

  Sí. Las instancias de DCS Redis 4.0 y 5.0 soportan la persistencia de AOF y de RDB. Puede realizar copias de respaldo de los datos en los archivos de RDB y de AOF en la consola y descargar los archivos.

5.4 ¿Por qué se eliminan con frecuencia los procesos durante la migración de datos?

Posible causa: La memoria es insuficiente.

Solución: expanda la memoria del servidor en el que se ejecuta el comando de migración.

5.5 ¿Dónde se almacenan los archivos de copia de respaldo de instancia de DCS? ¿Cómo se facturan?

Los archivos de copia de respaldo se almacenan en OBS. Actualmente, DCS y OBS no cobran por las copias de respaldo. En el futuro, se puede cobrar una cierta cantidad de tasas sobre la base de la norma unificada.

5.6 ¿Se migran todos los datos de una instancia de DCS Redis durante la migración en línea?

La migración entre instancias de nodo único y principal/en espera implica el conjunto completo de datos. Todas las bases de datos se migrarán y no podrá migrar las bases de datos especificadas. Después de la migración, una clave determinada permanecerá en la misma base de datos que estaba antes de la migración.

Por el contrario, una instancia de clúster solo tiene una base de datos, que es DB0. Durante la migración, se migran los datos en todas las ranuras de DB0.

5.7 ¿DCS soporta la persistencia de datos? ¿Cuál es el impacto de la persistencia?

Apoyo a la persistencia

- Instancias de DCS Redis:
  - Nodo único: no se admite la persistencia de datos.
  - Principal/en espera, separación de lectura/escritura, y clúster (excepto clústeres de réplica única): Se admite la persistencia de datos.
- Instancias de DCS Memcached:
  - Nodo único: no se admite la persistencia de datos.
  - Principal/en espera: Se admite la persistencia de datos.

Modos de persistencia

- DCS solo admite la persistencia de AOF de forma predeterminada. Puede habilitar o deshabilitar la persistencia según sea necesario. Todas las instancias, excepto las de nodo único y de clúster de réplica única, se crean con la persistencia de AOF habilitada.
- DCS no admite la persistencia de RDB de forma predeterminada y no puede configurar el parámetro **save**. Si se requiere persistencia de RDB para una instancia principal/en espera o de clúster de Redis 4.0 o posterior, puede utilizar la función de copia de respaldo y restauración para realizar una copia de respaldo de los datos de instancia en un archivo de RDB y almacenar los datos en OBS.

Disco utilizado para la persistencia

Para las instancias de DCS Redis 4.0 y posteriores, los datos se conservan en los discos de SSD.

Impacto de la persistencia de AOF

Después de activar la persistencia de AOF, el proceso Redis-Server necesita registrar las operaciones en el archivo de AOF para la persistencia de datos.

- Si el disco o la E/S del nodo informático subyacente es defectuoso, la latencia puede aumentar o puede producirse una conmutación principal/en espera.

- Redis-Server reescribe periódicamente el AOF. Durante una reescritura, la latencia puede ser alta durante un corto tiempo. Para obtener más información sobre las reglas de reescritura de AOF, consulte **¿Cuándo se activarán las reescrituras de AOF?**

Si las instancias de DCS se utilizan para acelerar las aplicaciones, se recomienda desactivar la persistencia para lograr un mayor rendimiento y estabilidad. Tenga cuidado al deshabilitar la persistencia. Sin persistencia, los datos almacenados en caché pueden perderse en los escenarios extremos (por ejemplo, cuando tanto el nodo principal como el nodo en espera son defectuosos).

Para deshabilitar la persistencia de AOF, establezca el parámetro **appendonly** en **no** en la página de detalles de la instancia.

5.8 ¿Cuándo se activarán las reescrituras de AOF?

Las reescrituras de AOF implican los siguientes conceptos:

- Reescritura de la ventana, que es actualmente de 01:00 a 04:59
- Umbral de uso del disco, que es del 50%

Las reescrituras de AOF se activan en los siguientes escenarios:

- Si el uso del disco alcanza el umbral (independientemente de si la hora actual está dentro de la ventana de reescritura), las reescrituras se activarán en instancias cuyo tamaño de archivo de AOF sea mayor que el tamaño del conjunto de datos de memoria.
- Si el uso del disco está por debajo del umbral y el tiempo actual está dentro de la ventana de reescritura, las reescrituras se activarán en instancias cuyo tamaño de archivo AOF sea mayor que la memoria del conjunto de datos multiplicada por 1.5.
- Si el uso del disco está por debajo del umbral pero el tiempo actual está fuera de la ventana de reescritura, las reescrituras se activarán en instancias cuyo tamaño de archivo de AOF sea mayor que la memoria máxima multiplicada por 4.5.

5.9 ¿Cuáles son las causas comunes de las fallas de migración de Redis?

- Compruebe si se produjo una conmutación principal/en espera durante la migración. Si ocurrió, póngase en contacto con el soporte técnico para deshabilitar temporalmente la conmutación principal/en espera hasta que se complete la migración.
- Para la migración en línea, compruebe si los comandos **SYNC** y **PSYNC** están deshabilitados en la instancia de Redis de origen. Si están deshabilitados, habilite para permitir la sincronización de datos.
- De forma predeterminada, una instancia de Clúster Proxy solo tiene una base de datos (DB0). Antes de migrar datos desde una instancia de nodo único o principal/en espera a una instancia de Clúster Proxy, compruebe si existen datos en bases de datos distintas de DB0. En caso afirmativo, habilite multi-BD para la instancia de Clúster Proxy haciendo referencia a la **Habilitación de Multi-BD**.
- De forma predeterminada, una instancia de Clúster Redis solo tiene una base de datos (DB0). Antes de migrar datos desde una instancia de nodo único o principal/en espera a una instancia de Clúster Redis, compruebe si existen datos en bases de datos distintas de DB0. Para garantizar que la migración tenga éxito, mueva todos los datos a DB0 consultando **Migración en línea con Rump**.

5.10 ¿Puedo migrar datos a varias instancias de destino en una tarea de migración?

No. Una tarea de migración permite migrar datos a solo una instancia de destino. Para migrar datos a varias instancias de destino, cree varias tareas de migración.

5.11 ¿Cómo puedo activar los comandos SYNC y de PSYNC?

- Migración dentro de DCS:
  - De forma predeterminada, los comandos **SYNC** y **PSYNC** se pueden utilizar cuando Redis autohospedado se migra a DCS.
  - Durante la migración en línea entre instancias de DCS Redis en la misma región bajo la misma cuenta, los comandos **SYNC** y **PSYNC** se activan automáticamente.
  - Durante la migración en línea entre las instancias de DCS Redis en diferentes regiones o bajo diferentes cuentas dentro de una región, los comandos **SYNC** y **PSYNC** no se habilitan automáticamente y no se puede usar la migración en línea. Puede migrar datos mediante los archivos de copia de respaldo.
- Migración de otros proveedores de nube a DCS:
  - Por lo general, los proveedores de nube deshabilitan los comandos **SYNC** y **PSYNC**. Si desea utilizar la función de migración en línea en la consola de DCS, póngase en contacto con el personal de O&M del proveedor de la nube de origen para habilitar los comandos. Para la migración sin conexión, puede importar los archivos de copia de respaldo.
  - Si no se requiere una migración incremental, puede realizar una migración completa haciendo referencia a **Migración completa en línea de Redis desde otra nube con redis-shake**. Este método no depende de **SYNC** y **PSYNC**.

5.12 ¿Por qué falla la creación de tareas de migración?

Causas posibles:

1. Los recursos subyacentes son insuficientes.
2. Las especificaciones del ECS utilizadas para la migración son insuficientes.
3. La memoria del Redis de destino creado antes de la migración es menor que la del Redis de origen.

5.13 ¿Se sobrescribirán las mismas claves durante la migración de datos o la importación de copias de respaldo?

Si los datos existen tanto en las instancias de origen como en las de destino, los datos de destino se sobrescriben mediante los datos de origen. Si los datos solo existen en la instancia de destino, los datos se conservarán.

La incoherencia entre los datos de origen y de destino después de la migración puede deberse a los datos de destino que existían y se retuvieron antes de la migración.

6 Análisis de claves grandes, análisis de claves de mucho uso y escaneo de claves caducadas

6.1 ¿Qué son las claves grandes y las claves de mucho uso?

Término	Definiciones
Clave grande	Hay dos tipos de las claves grandes: ● Claves que tienen un valor grande, por ejemplo, una clave String de 10 MB o una clave Hash, List o Set de 100 MB (todos los conceptos combinados). Si el tamaño de una sola clave String supera los 10 KB, o si el tamaño de todos los conceptos de una clave combinada supera los 50 MB, la clave se define como una clave grande. ● Las claves que tienen un gran número de conceptos, por ejemplo, una clave Hash que tiene 10,000 conceptos. Si el número de conceptos en una clave supera los 5000, la clave se define como una clave grande.
Clave de mucho uso	Una clave se define como una clave de mucho uso si se solicita con frecuencia o si ocupa un gran número de recursos. Por ejemplo: ● En una instancia de clúster, una partición procesa 10,000 solicitudes por segundo, entre las cuales 3000 se realizan en la misma clave. ● En una instancia de clúster, una partición utiliza un total de 100 Mbits/s de ancho de banda entrante y saliente, entre los que la operación **HGETALL** utiliza 80 Mbits/s en una clave de hash.

6.2 ¿Cuál es el impacto de las claves grandes o de las claves de mucho uso?

Categoría	Impacto
Clave grande	**Las especificaciones de instancia no se pueden modificar.** La modificación de la especificación de una instancia de Clúster Redis implica un reequilibrio (migración de datos entre los nodos). Redis tiene un límite en la migración de claves. Si la instancia tiene una sola clave superior a 512 MB, la modificación fallará cuando se agote el tiempo de migración de clave grande entre nodos. Cuanto más grande sea la clave, más probabilidades hay de que la migración falle.
	Se falla la migración de datos. Durante la migración de datos, si una clave tiene muchos conceptos, otras claves se bloquearán y se almacenarán en el búfer de memoria del ECS de migración. Si se bloquean durante mucho tiempo, la migración fallará.
	Particiones de clúster están desequilibradas. ● El uso de memoria de las particiones está desequilibrado. Por ejemplo, si una partición utiliza una memoria grande o incluso utiliza la memoria, las claves de esta partición se desalojan y los recursos de otras particiones se desperdician. ● El uso de ancho de banda de las particiones está desequilibrado. Por ejemplo, el control de flujo se activa repetidamente en una partición.
	Aumenta la latencia de la ejecución de comandos del cliente. Las operaciones lentas en una clave grande bloquean otros comandos, lo que resulta en un gran número de las consultas lentas.
	El control de flujo se activa en la instancia. La lectura frecuente de datos de claves grandes agota el ancho de banda saliente de la instancia, activando el control de flujo. Como resultado, se produce un gran número de comandos de tiempo de espera o consultas lentas, lo que afecta a los servicios.
	Se activa la conmutación principal/en espera. Si la operación de **DEL** de alto riesgo se realiza en una clave grande, el nodo principal puede bloquearse durante mucho tiempo, provocando una conmutación principal/en espera.
Clave de mucho uso	**Particiones de clúster están desequilibradas.** Si solo la partición donde se encuentra la clave de mucho uso está ocupado procesando las consultas de servicio, puede haber cuellos de botella de rendimiento en la partición única y los recursos informáticos de otras particiones pueden desperdiciarse.

Categoría	Impacto
	Oleadas de uso de la CPU. Un gran número de operaciones en las claves de mucho uso puede causar un uso alto de la CPU. Si las operaciones se realizan en una sola partición de clúster, el uso de CPU de la partición donde se encuentra la clave de mucho uso aumentará. Esto ralentizará otras solicitudes y el rendimiento general. Si el volumen de servicio aumenta bruscamente, se puede activar una conmutación principal/en espera.
	Puede producirse una avería de la caché. Si Redis no puede manejar la presión sobre las claves de mucho uso, las solicitudes llegarán a la base de datos. La base de datos puede descomponerse a medida que su carga aumenta drásticamente, afectando a otros servicios.

6.3 ¿Cómo puedo evitar las claves grandes y las claves de mucho uso?

- **Mantenga el tamaño de las cadenas dentro de los 10 KB** y **la cantidad** de Hashes, Lists, Sets o Zsets **dentro de los 5000**.
- Al nombrar claves, utilice la abreviatura del nombre del servicio como prefijo y no utilice caracteres especiales como espacios, frenos de línea, comillas simples o dobles y otros caracteres de escape.
- No confíe demasiado en las transacciones de Redis.
- El rendimiento de las conexiones cortas es pobre. Utilice clientes con los grupos de conexiones.
- No habilite la persistencia de datos si utiliza Redis solo para el almacenamiento en caché y puede tolerar las pérdidas de datos.
- Para obtener más información sobre cómo optimizar las claves grandes y las claves de mucho uso, consulte la tabla siguiente.

Categoría	Método
Clave grande	**Dividir las claves grandes.** Escenarios: ● **Si la clave grande es una String**, puede dividirla en varios pares de clave-valor y usar **MGET** o una canalización que consiste en varias operaciones de **GET** para obtener los valores. De esta manera, se puede dividir la presión de una única operación. Para un ejemplo de grupo, la presión de operación puede distribuirse uniformemente a múltiples particiones, reduciendo el impacto en una sola partición. ● **Si la clave grande contiene varios conceptos y tienen que operarse juntos**, la clave grande no se puede dividir. Puede quitar la clave grande de Redis y almacenarla en otros medios de almacenamiento en su lugar. Este escenario debe ser evitado por el diseño. ● **Si la clave grande contiene varios conceptos, y solo se operan algunos conceptos cada vez**, separe los conceptos. Tome una clave Hash como ejemplo. Cada vez que ejecuta el comando **HGET** o **HSET**, el resultado del módulo de valor hash *N* (personalizado en el cliente) determina en qué clave cae el campo. Este algoritmo es similar al utilizado para calcular las ranuras en Clúster Redis.
	Almacenar las claves grandes en otros medios de almacenamiento. Si una clave grande no se puede dividir, no es adecuada para ser almacenada en Redis. Puede almacenarlo en otros medios de almacenamiento, como **SFS** u otras bases de datos de NoSQL, y eliminar la clave grande de Redis. **ATENCIÓN** No utilice el comando **DEL** para eliminar las claves grandes. De lo contrario, Redis puede bloquearse o incluso puede producirse una conmutación principal/en espera.
	Establecer la caducidad adecuada y eliminar los datos caducados con regularidad. La expiración apropiada impide que los datos caducados permanezcan en Redis. Debido a que Redis está libre de perezosos, los datos caducados no se pueden eliminar a tiempo. Si esto ocurre, **escanee las claves caducadas**.
Clave de mucho uso	**Dividir las solicitudes de lectura y de escritura.** Si se lee con frecuencia una clave de mucho uso, **configure la separación de lectura/escritura** en el cliente para reducir el impacto en el nodo principal. También puede agregar réplicas para cumplir con los requisitos de lectura, pero no puede haber demasiadas réplicas. En DCS, las réplicas replican datos del principal en paralelo. Las réplicas son independientes entre sí y el retardo de replicación es corto. Sin embargo, si hay un gran número de réplicas, el uso de CPU y la carga de red en el nodo principal serán altas.

Categoría	Método
	Utilizar la caché del cliente o la caché local. Si sabe qué claves se utilizan con frecuencia, puede diseñar una arquitectura de caché de dos niveles (caché de cliente/local y Redis remoto). Los datos utilizados con frecuencia se obtienen en primer lugar de la caché local. La caché local y la caché remota se actualizan con escrituras de datos al mismo tiempo. De esta manera, la presión de lectura sobre los datos a los que se accede con frecuencia puede separarse. Este método es costoso porque requiere cambios en la arquitectura y el código del cliente.
	Diseñar un interruptor o mecanismo de degradación. Las claves de mucho uso pueden resultar fácilmente en la ruptura de la caché. Durante las horas pico, las solicitudes se pasan a la base de datos de backend, causando avalancha de servicio. Para asegurar la disponibilidad, el sistema debe tener un interruptor o mecanismo de degradación para limitar el tráfico y degradar los servicios si se produce una avería.

6.4 ¿Cómo analizo las claves de mucho uso de una instancia de DCS Redis 3.0?

DCS for Redis 3.0 no admite el análisis de clave de mucho uso en la consola. También puede utilizar los siguientes métodos para analizar las claves de mucho uso:

- Método 1: Analice la estructura del servicio y el despliegue del servicio para descubrir las posibles claves de mucho uso.

  Por ejemplo, las claves de mucho uso se pueden encontrar fácilmente en el código de servicio durante las ventas flash o los inicios de sesión de los usuarios.

  Ventaja: Simple y fácil de desplegar.

  Desventaja: Requiere familiaridad con el código de servicio. Además, el análisis se vuelve más difícil a medida que los escenarios de servicio se vuelven más complejos.
- Método 2: Recopile estadísticas de acceso a claves en el código del cliente para descubrir las claves de mucho uso.

  Desventaja: Requiere la modificación intrusiva del código.
- Método 3: Capture y analice los paquetes.

  Ventaja: Simple y fácil de desplegar.

6.5 ¿Cómo puedo detectar claves grandes y claves de mucho uso por adelantado?

Método	Descripción
Con **Big Key Analysis** y **Hot Key Analysis** en la consola de DCS	Consulte **Análisis de claves grande y de claves de mucho uso**.
Con las opciones **bigkeys** y **hotkeys** en redis-cli	● redis-cli utiliza la opción **bigkeys** para recorrer todas las claves de una instancia de Redis y devuelve las estadísticas de clave generales y la clave más grande de seis tipos de datos: Strings, Lists, Hashes, Sets, Zsets y Streams. El comando es **redis-cli -h** *<Dirección de conexión de instancia>* **-p** *<Número de puertos>* **-a** *<Contraseña>* **--bigkeys**. ● En Redis 4.0 y las versiones posteriores, puede usar la opción **hotkeys** para encontrar rápidamente las claves de mucho uso en redis-cli. Ejecute este comando durante la ejecución del servicio para encontrar las claves de mucho uso **redis-cli -h** *<Dirección de conexión de instancia>* **-p** *<Número de puertos>* **-a** *<Contraseña>* **--hotkeys**. Los detalles de la clave de mucho uso se pueden obtener de la parte de resumen en el resultado devuelto.
Búsqueda de las claves grandes con los comandos de Redis	Si hay un patrón de las claves grandes, por ejemplo, el prefijo es de **cloud:msg:test**, puede usar un programa para buscar claves que coincidan con el prefijo y, a continuación, ejecutar comandos para consultar el número de miembros de la clave y consultar los tamaños de las claves para buscar las grandes. ● Comandos para consultar el número de miembros: **LLEN**, **HLEN**, **XLEN**, **ZCARD**, **SCARD** ● Comandos para consultar el uso de memoria de las clavs: **DEBUG OBJECT**, **MEMORY USAGE** **ATENCIÓN** Este método consume un gran número de los recursos informáticos. Para garantizar la ejecución del servicio, no utilice este método en instancias con una gran presión de servicio.

Método	Descripción
Búsqueda de las claves grandes con redis-rdb-tools	**redis-rdb-tools** es una herramienta de código abierto para analizar los archivos de instantáneas de RDB de Redis. Puede utilizarlo para analizar el uso de memoria de todas las claves de una instancia de Redis. Para utilizar este método, debe **exportar el archivo RDB de una instancia** en la página **Backups & Restorations** de la consola de DCS. **ATENCIÓN** Este método no afecta a la ejecución del servicio, pero no es tan oportuno como el análisis en línea.

Las instancias de DCS Redis 3.0 no admiten el análisis de claves de mucho uso. Sin embargo, puede **configurar alarmas** para detectar claves de mucho uso.

- Configurar reglas de alarma para la métrica **Memory Usage** de los nodos de instancia.

  Si un nodo tiene una clave grande, el uso de memoria del nodo es mucho mayor que el de otros nodos. En este caso, se activa una alarma para ayudarle a encontrar la clave potencialmente problemática.
- Configure reglas de alarma para las métricas **Maximum Inbound Bandwidth**, **Maximum Outbound Bandwidth** y **CPU Usage** de los nodos de instancia.

  Si un nodo tiene una clave de mucho uso, el ancho de banda y el uso de CPU del nodo son mucho más altos que los de otros nodos. En este caso, se activa una alarma para ayudarle a encontrar la clave potencialmente problemática.

6.6 ¿Cómo elimina DCS las claves caducadas?

Pregunta

¿Cuáles son las reglas para la eliminación programada de las claves caducadas a diario? ¿Puedo personalizar las reglas?

Mecanismos para eliminar las claves caducadas

- Borrado libre perezoso: La estrategia de borrado se controla en el bucle principal de eventos de E/S. Antes de ejecutar una orden de lectura/escritura, se llama a una función para comprobar si la clave a la que se accede ha expirado. Si ha caducado, se eliminará y se devolverá una respuesta indicando que la clave no existe. Si la clave no ha caducado, se reanuda la ejecución del comando.
- Eliminación programada: Una función de evento de tiempo se ejecuta en ciertos intervalos. Cada vez que se ejecuta la función, se comprueba una colección aleatoria de claves y se eliminan las claves caducadas.

  NOTA

  Para evitar bloqueos prolongados en el subproceso principal de Redis, no todas las claves se comprueban en cada evento de tiempo. En su lugar, una colección aleatoria de claves se comprueba cada vez. Como resultado, la memoria utilizada por las claves caducadas no se puede liberar rápidamente.

Soluciones

- Configure las tareas programadas de análisis de claves de mucho uso haciendo referencia a **Análisis de claves de mucho uso** o utilice el comando **SCAN** para recorrer todas las claves de forma programada y quitar las claves caducadas de la memoria.
- Configure una tarea programada para analizar todos los nodos principales de la instancia. Todas las claves serán escaneadas, y Redis determinará si las claves han caducado. Las claves vencidas serán liberadas. Para obtener más información, consulte la sección **Escaneo de las claves caducadas**.

6.7 ¿Cuánto tiempo se almacenan las claves? ¿Cómo configuro la caducidad de la clave?

- Duración del almacenamiento de claves
  - Las claves que no tienen una caducidad se almacenan permanentemente.
  - Las claves que tienen una caducidad se eliminan después de que caduquen. Para obtener más información, consulte la sección **Escaneo de las claves caducadas**.
  - Para eliminar el conjunto de caducidad de una clave, ejecute el comando **PERSIST**.
- Ajuste de la expiración de la clave

  Puede ejecutar el comando **EXPIRE** o **PEXPIRE** para establecer el tiempo de caducidad de la clave. Por ejemplo, si ejecuta **expire key1 100**, key1 caducará en 100 segundos. Si ejecuta **pexpire key2 1800**, key2 caducará en 1800 milisegundos.

  EXPIRE establece la caducidad de la clave en segundos, y **PEXPIRE** establece la caducidad de la clave en milisegundos.

7 Comandos de Redis

7.1 ¿Cómo puedo borrar los datos de Redis?

Tenga cuidado al borrar los datos.

- Redis 3.0

  Los datos de una instancia de DCS Redis 3.0 no se pueden borrar en la consola y solo se pueden borrar mediante el comando **FLUSHDB** o **FLUSHALL** en redis-cli.

  Ejecute el comando **FLUSHALL** para borrar todos los datos de la instancia.

  Ejecute el comando **FLUSHDB** para borrar los datos de la base de datos seleccionada actualmente.

- Redis 4.0 o posterior

  Para borrar datos de una instancia de DCS Redis 4.0 o posterior, puede ejecutar el comando **FLUSHDB** o **FLUSHALL** en redis-cli, usar la función de borrado de datos en la consola de DCS o ejecutar el comando **FLUSHDB** en la Web CLI.

  Para borrar datos de una instancia de Clúster Redis, ejecute el comando **FLUSHDB** o **FLUSHALL** en cada partición de la instancia. De lo contrario, es posible que los datos no se borren completamente.

  NOTA

  - Actualmente, la función de borrado de datos y el acceso de Web CLI en la consola solo son compatibles con instancias de DCS Redis 4.0 o posteriores.
  - Cuando ejecuta el comando **FLUSHDB** en Web CLI, solo se borra una partición a la vez. Si hay varias particiones, conéctese a cada maestra y ejecute el comando **FLUSHDB** en cada maestra.
  - Los datos de Clúster Redis no se pueden borrar mediante el uso de Web CLI.

7.2 ¿Cómo encuentro las claves especificadas y recorro todas las claves?

Búsqueda de las claves especificadas

El análisis de los claves grandes y claves de mucho uso no admite la búsqueda de claves con las condiciones especificadas. Para buscar claves con el prefijo o sufijo especificado, utilice el comando **SCAN**.

Por ejemplo, para buscar claves que contengan la letra *a* en una instancia de Redis, ejecute el siguiente comando en redis-cli:

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

Recorriendo todas las claves

No utilice el comando **KEYS** para recorrer todas las claves de una instancia porque el comando **KEYS** es complejo y puede bloquear Redis. Para recorrer todas las claves de una instancia de DCS Redis, ejecute el siguiente comando en redis-cli:

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

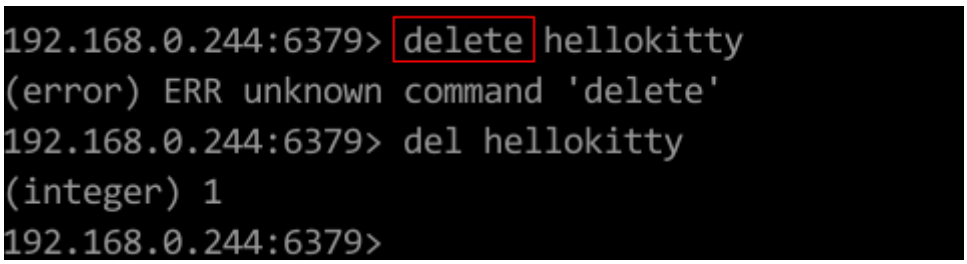
Para obtener más información sobre el comando **SCAN**, visite el **sitio web oficial de Redis**.

7.3 ¿Por qué no puedo ejecutar algunos comandos de Redis?

Las causas posibles incluyen las siguientes:

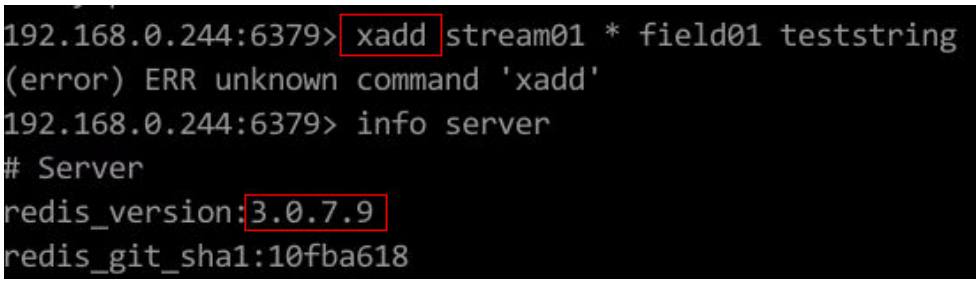
- El comando se escribe incorrectamente.

  Como se muestra en la siguiente figura, se devuelve el mensaje de error porque el comando correcto para eliminar una clave debe ser **del**.

- Un comando disponible en una versión de Redis superior se ejecuta en una versión de Redis inferior.

  Como se muestra en la siguiente figura, el mensaje de error se devuelve porque se ejecuta un comando stream (disponible en Redis 5.0) en Redis 3.0.



- El comando está deshabilitado en DCS.

  Por motivos de seguridad, algunos comandos de Redis están deshabilitados en DCS. Para obtener más información sobre los comandos de Redis deshabilitados y restringidos, consulte la sección **Compatibilidad con el comando** .

- El comando no se puede ejecutar en la Web CLI.

  Además de los comandos Redis deshabilitados y restringidos, los comandos **KEYS** también están restringidos en la Web CLI.

- El script de LUA no se puede ejecutar.

  Por ejemplo, el mensaje de error "ERR unknown command 'EVAL'" indica que su instancia de DCS Redis es de una versión inferior que no admite el script de LUA. En este caso, envíe un ticket de servicio para la instancia a actualizar.

- Los comandos **CLIENT SETNAME** y **CLIENT GETNAME** no se pueden ejecutar.

  La instancia de DCS Redis es de una versión inferior que no admite estos comandos. En este caso, envíe un ticket de servicio para la instancia a actualizar.

- Los siguientes comandos están deshabilitados para las instancias de DCS Redis de **cluster** creadas antes del 10 de julio de 2018. Puede actualizar dicha instancia enviando un ticket de servicio.

  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 qué se devuelve "permission denied" cuando ejecuto el comando Keys en Web CLI?

El comando **KEYS** está deshabilitado en la Web CLI. Este comando solo se puede ejecutar en redis-cli.

7.5 ¿Cómo cambio el nombre de los comandos de alto riesgo?

Actualmente, solo puede cambiar el nombre de los siguientes comandos críticos: **COMMAND**, **KEYS**, **FLUSHDB**, **FLUSHALL**, **HGETALL**, **SCAN**, **HSCAN**, **SSCAN** y **ZSCAN** para las instancias de DCS Redis 4.0 y 5.0. Para obtener más información, consulte la sección **Comandos de cambio de nombre**.

NOTA

- Actualmente, Redis no admite la desactivación de comandos. Para evitar riesgos al utilizar los comandos anteriores, cámbielos el nombre. Para obtener más información sobre los comandos admitidos y deshabilitados en DCS, consulte **Compatibilidad de comandos**.
- Los nuevos nombres de comandos surtirán efecto solo después de reiniciar la instancia. Recuerde los nuevos nombres de comandos porque no se mostrarán en la consola por motivos de seguridad.

7.6 ¿DCS for Redis soporta la canalización?

Sí.

Para las instancias de DCS Redis 4.0 y 5.0 en el modo de Clúster Redis, asegúrese de que todos los comandos de una canalización se ejecuten en la misma partición.

7.7 ¿DCS for Redis soporta los comandos INCR y EXPIRE?

Sí.

Para obtener más información acerca de la compatibilidad de comandos de Redis, vea **Compatibilidad de comandos de Redis**.

7.8 ¿Por qué un comando de Redis no tiene efecto?

Ejecute el comando en redis-cli para comprobar si el comando tiene efecto.

A continuación se describen dos escenarios:

- Escenario 1: Establecer y consultar el valor de una clave para comprobar si se funcionan los comandos **SET** y **GET**.

  El comando **SET** se utiliza para establecer el valor de cadena. Si el valor no se cambia, ejecute los siguientes comandos en redis-cli para acceder a la instancia:

  ```
  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>
  ```

- Escenario 2: Si el tiempo de espera establecido con el comando **EXPIRE** es incorrecto, realice las siguientes operaciones:

  Establezca el tiempo de espera en 10 segundos y ejecute el comando **TTL** para ver el tiempo restante. Como se muestra en el siguiente ejemplo, el tiempo restante es 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

Los clientes de Redis (incluidos redis-cli, clientes de Jedis y clientes de Python) se comunican con el servidor de Redis usando un protocolo binario.

Si los comandos de Redis se ejecutan correctamente en redis-cli, el problema puede estar en el código de servicio. En este caso, cree logs en el código para su posterior análisis.

7.9 ¿Hay un límite de tiempo en la ejecución de comandos de Redis? ¿Qué sucederá si un comando se agota?

Los tiempos de espera de los comandos Redis se pueden controlar en el extremo del cliente o del servidor.

- Los tiempos de espera en el extremo del cliente se controlan en el código del cliente. Puede determinar los tiempos de espera que se adapten a las necesidades de servicio. Por ejemplo, si utiliza Lettuce, un cliente Java, configure el parámetro **timeout**.
- En el extremo del servidor, el parámetro **timeout** se establece en **0** de forma predeterminada, lo que indica que las conexiones nunca se terminarán. Modifique la configuración del parámetro haciendo referencia a **Modificación de parámetros de configuración**.

7.10 ¿Puedo configurar las claves de Redis para que no distingan entre mayúsculas y minúsculas?

No. Al igual que en Redis de código abierto, las claves en DCS for Redis distinguen entre mayúsculas y minúsculas y no se pueden configurar para que sean insensibles entre mayúsculas y minúsculas.

7.11 ¿Puedo ver los comandos de Redis más utilizados?

No. Redis no graba comandos y no admite la visualización de los comandos más utilizados.

7.12 Errores comunes de Web CLI

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

   Este error indica que el comando de Redis ejecutado tiene un error de parámetro (error de sintaxis). Vuelva a escribir el comando haciendo referencia al protocolo de comando de Redis de código abierto.

2. ERR unknown command 'xxx'

   Este error indica que el comando es desconocido o no es un comando válido definido por Redis. Vuelva a escribir el comando haciendo referencia al protocolo de comando de Redis de código abierto.

3. ERR Unsupported command: 'xxx'

   Este error indica que el comando está deshabilitado para las instancias de DCS Redis. Para obtener más información, consulte los **Comandos de Web CLI**.

8 Monitoreo y alarma

8.1 ¿Cómo puedo ver las conexiones simultáneas actuales y las conexiones máximas de una instancia de DCS Redis?

Consulta de conexiones simultáneas de una instancia de DCS Redis

El número de conexiones en tiempo real recibidas por una instancia de DCS es una métrica que se puede ver en la consola. Para obtener detalles sobre cómo ver las métricas, consulte la sección **Consulta de métricas de supervisión de DCS** .

En la consola de Cloud Eye, busca la métrica **Connected Clients**. Haga clic en  para ver los detalles de supervisión en un gráfico ampliado.

Especifique un intervalo de tiempo para ver la métrica en un período de supervisión específico. Por ejemplo, puede seleccionar un período de 10 minutos para ver el número de conexiones recibidas durante el período. En el gráfico, puede ver la tendencia y el número total de conexiones recibidas durante el período.

En la consola de Cloud Eye, también puede ver otras métricas de supervisión de sus instancias de DCS, por ejemplo:

- Uso de CPU
- Uso de memoria
- Memoria utilizada
- Operaciones por segundo

Consulta o modificación del máximo de conexiones de una instancia

Puede ver el número predeterminado y el número máximo permitido de conexiones en la página de creación de instancia o en el **documento**.

Después de crear una instancia, puede ver o cambiar el valor de **maxclients** (el número máximo de conexiones) en la página de la instancia **Instance Configuration** > **Parameters**. (Las instancias de Clúster Proxy no tienen este parámetro.)

Si se excede el límite, las solicitudes de exceso serán rechazadas y las conexiones se agotarán.

8.2 ¿DCS for Redis soporta las auditorías de comandos?

No. Para garantizar las operaciones de lectura y escritura de alto rendimiento, DCS for Redis no audita los comandos. Los comandos no se imprimen.

8.3 ¿Qué debo hacer si los datos de supervisión de una instancia de DCS Redis son anormales?

Si tiene alguna duda sobre los datos de monitoreo de una instancia de DCS Redis, puede acceder a la instancia a través de redis-cli y ejecutar el comando **INFO ALL** para ver las métricas. Para obtener más información sobre la salida del comando **INFO ALL**, consulte **http://www.redis.io/commands/info**.

8.4 ¿Por qué la memoria usada es mayor que la memoria disponible?

Para las instancias de DCS de nodo único y principal/en espera, la memoria de instancia utilizada se mide mediante el proceso del servidor Redis. Para las instancias de DCS de clúster, la memoria de clúster utilizada es la suma de la memoria utilizada de todos los fragmentos del clúster.

Debido al despliegue interna del servidor redis de código abierto, la memoria de instancia usada es normalmente ligeramente más alta que la memoria de instancia disponible.

Redis asigna memoria usando zmalloc. No comprueba si used_memory excede max_memory cada vez que se asigna la memoria. En su lugar, comprueba si la used_memory actual excede max_memory al comienzo de una tarea periódica o procesamiento de órdenes. Si used_memory excede max_memory, se activa el desalojo. Por lo tanto, las restricciones de la política max_memory no se despliegan en tiempo real o rígidamente. Un caso en el que la used_memory es mayor que la max_memory puede ocurrir ocasionalmente.

8.5 ¿Por qué el uso del ancho de banda supera el 100%?

La información básica sobre la métrica de uso de ancho de banda es la siguiente.

ID de la métrica	Nombre de la métrica	Descripción	Rango de valores	Objeto y dimensión supervisados	Período de monitoreo (datos brutos)
bandwidth_usage	Bandwidth Usage	Porcentaje del ancho de banda utilizado al límite máximo de ancho de banda	0–200%	Objeto supervisado: Instancias principal/en espera de DCS Redis 4.0 o 5.0 Servidor de Redis de la instancia de Clúster Redis de DCS Redis 4.0 o 5.0 Dimensión: dcs_cluster_node	1 minuto

Uso de ancho de banda = (flujo de entrada + flujo de salida)/(2 x ancho de banda máximo) x 100%

De acuerdo con la fórmula, el uso del ancho de banda cuenta en el flujo de entrada y el flujo de salida, que incluyen el tráfico para la replicación entre las principales y las réplicas. Por lo tanto, el tráfico total es mayor que el tráfico de servicio normal.

Si el valor de la métrica **Flow Control Times** es mayor que 0, se ha alcanzado el ancho de banda máximo y se ha realizado el control de flujo.

Sin embargo, las decisiones de control de flujo se toman sin considerar el tráfico para la replicación entre las principales y las réplicas. Por lo tanto, a veces el uso de ancho de banda excede el 100%, pero el número de tiempos de control de flujo es 0.

8.6 ¿Por qué se muestra la métrica de conexiones rechazadas?

Si se muestra la métrica **Rejected Connections**, compruebe si el número de clientes conectados excede el número máximo permitido de conexiones de las instancias.

- Para comprobar el número máximo permitido de conexiones, vaya a la pestaña **Parameters** de la instancia y compruebe el valor del parámetro **maxclients**. (Las instancias de Proxy Clúster no tienen este parámetro. Puede ver el número máximo de conexiones en la página de creación de instancia.)
- Para comprobar el número actual de conexiones, vaya a la página de pestaña **Performance Monitoring** de la instancia y compruebe la métrica **Connected Clients**.

Si el número actual de conexiones alcanza el límite superior, puede ajustar el valor de **maxclients**. Si el valor de **maxclients** ya no se puede aumentar, aumente las especificaciones de la instancia.

8.7 ¿Por qué se activa el control de flujo? ¿Cómo lo manejo?

El control de flujo se activa cuando el tráfico utilizado por una instancia de Redis en un período excede el ancho de banda máximo. Las conexiones pueden descartarse debido al control de flujo, lo que resulta en una alta latencia de servicio y excepciones de conexión de cliente.

NOTA

Para obtener detalles sobre el ancho de banda máximo permitido, consulte la columna "Assured/Maximum Bandwidth" de diferentes tipos de instancia enumerados en las **Especificaciones de instancias de DCS**.

Incluso si el uso de ancho de banda es bajo, el control de flujo todavía puede activarse. El uso del ancho de banda en tiempo real se notifica una vez en cada período de informe. Los controles de flujo se comprueban cada segundo. El tráfico puede aumentar en segundos y luego retroceder entre períodos de informes. En el memento en que se informa del uso del ancho de banda, es posible que ya se haya restaurado al nivel normal.

Para instancias principal/en espera:

- Si el control de flujo siempre se activa cuando el uso del ancho de banda es bajo, puede haber microrráfagas de servicio o las claves grandes o de mucho uso. En este caso, compruebe si hay claves grandes o de mucho uso.
- Si el uso de ancho de banda sigue siendo alto, se puede exceder el límite de ancho de banda. En este caso, ampliar la capacidad. Una mayor capacidad admite un mayor ancho de banda.

Para instancias de clúster:

- Si el control de flujo se activa solo en una o unas particiones, ellas pueden tener las claves grandes o de mucho uso.
- Si el control de flujo o el uso de ancho de banda alto se produce en todas o en la mayoría de las particiones al mismo tiempo, el uso de ancho de banda de la instancia ha alcanzado el límite. En este caso, expanda la capacidad de la instancia.

NOTA

- Realice análisis de las claves grandes o de mucho uso en la consola de DCS y tome las medidas correspondientes. Para obtener más información, consulte **Análisis de claves grandes o de claves de mucho uso**.
- La ejecución de comandos (como **KEYS**) que consumen muchos recursos puede provocar un uso elevado de CPU y de ancho de banda. Como resultado, se activa el control de flujo.

9 Conmutación entre principal/en espera

9.1 ¿Cuándo se produce una conmutación principal/en espera?

Una conmutación principal/en espera puede ocurrir en los siguientes escenarios:

- Se inicia una operación de conmutación principal/en espera en la consola de DCS.
- Una conmutación principal/en espera se activará cuando el nodo principal de una instancia principal/en espera falle.

  Por ejemplo, si se usan comandos (como **KEYS**) que consumen muchos recursos o los logs se envejecen y se eliminan por lotes, el uso de la CPU aumentará, activando una conmutación principal/en espera.
- Si reinicia una instancia principal/en espera en la consola de DCS, se activará una conmutación principal/en espera.

Después de que se produzca una conmutación principal/en espera, recibirá una notificación. Compruebe si los servicios de cliente se están ejecutando correctamente. Si no es así, compruebe si la conexión de TCP es normal y si se puede restablecer después de la conmutación principal/en espera para restaurar los servicios.

9.2 ¿Cómo afecta la conmutación principal/e espera a los servicios?

Si se produce un error en una instancia de DCS principal/en espera o de clúster, se activa automáticamente una conmutación por error. Los servicios pueden interrumpirse durante menos de medio minuto durante la detección de excepciones y la conmutación por error.

9.3 ¿Necesita el cliente cambiar la dirección de conexión después de una conmutación principal/en espera?

No. Si el nodo principal falla o se realiza una conmutación principal/en espera, el nodo en espera será promovido a principal y tomará la dirección IP original.

9.4 ¿Cómo funciona la replicación de Redis principal/en espera?

Las instancias principal/en espera de Redis también se denominan las instancias maestra/esclava. Generalmente, las actualizaciones del nodo de caché principal se replican automáticamente y asincrónicamente al nodo de caché en espera. Esto significa que los datos en el nodo de caché en espera pueden no ser siempre coherentes con los datos en el nodo de caché principal. La inconsistencia se ve típicamente cuando la velocidad de escritura de E/S del nodo principal es más rápida que la velocidad de sincronización del nodo de espera o se produce una latencia de red entre los nodos principal y en espera. Si se produce una conmutación por error cuando algunos datos aún no se replican en el nodo en espera, dichos datos pueden perderse después de la conmutación por error.

10 Compras y permisos

10.1 ¿Por qué no puedo crear una instancia de DCS Redis o Memcached?

- La subred no tiene suficientes direcciones IP.

  Análisis: a cada nodo de una instancia de DCS se le debe asignar una dirección IP. Por lo tanto, una instancia de nodo único requiere una dirección IP, una instancia principal/en espera requiere dos direcciones IP y una instancia de clúster requiere múltiples direcciones IP.

  Solución: cree la instancia en una subred diferente dentro de la VPC o libere direcciones IP en la subred actual.

- El usuario de IAM no tiene los permisos necesarios para crear una instancia.

  Análisis: al grupo al que pertenece el usuario se le debe conceder la política **DCS FullAccess** o el rol **DCS Administrator** u otras políticas que contengan los permisos necesarios para crear las instancias de DCS.

  Solución: cree una instancia de DCS como administrador.

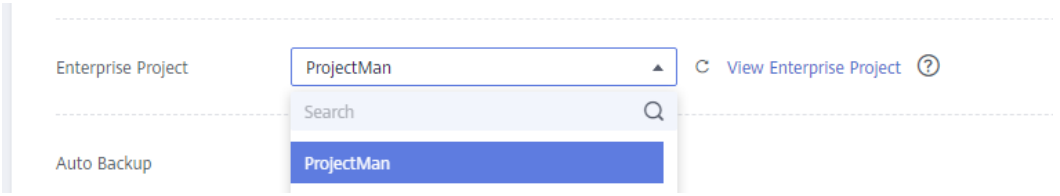
10.2 ¿Por qué no puedo ver la información de subred y el grupo de seguridad al crear una instancia de DCS?

Esto puede deberse a que no tiene los roles **Server Administrator** y **VPC Administrator**. Para obtener más información sobre cómo agregar permisos de usuario, consulte **Modificación de permisos de grupo de usuarios**.

10.3 ¿Por qué no puedo seleccionar el proyecto de empresa requerido al crear una instancia de DCS?

Síntomas

El proyecto de empresa deseado no se muestra durante la creación de instancias.



Motivo

El grupo de usuarios no tiene permisos de DCS en el proyecto de empresa deseado.

Solución

1. Inicie sesión en la consola de DCS.
2. En la esquina superior derecha, elija **Enterprise** >**Project Management**. En la página mostrada, haga clic en **View Resource** en la fila que contiene el proyecto de empresa deseado.
3. Haga clic en la ficha **Permissions**. A continuación, haga clic en **Authorize User Group**.

   NOTA

   Haga clic en **Authorize User Group** para conceder permisos a un grupo de usuarios o haga clic en **Authorize User** para conceder permisos a un usuario.
4. Haga clic en **Authorize** en la fila que contiene el usuario o grupo de usuarios al que desea conceder los permisos.
5. Busque y seleccione la política **DCS FullAccess**, haga clic en **Next** y haga clic en **OK**.

   Para obtener más información acerca de las políticas de permisos de DCS, vea **Gestión de permisos**.

10.4 ¿Por qué un usuario de IAM no puede ver una nueva instancia de DCS Redis?

Síntomas

Un usuario de IAM no puede ver una instancia de DCS Redis recién adquirida.

Causa posible

El usuario de IAM no tiene permisos para el proyecto de empresa al que pertenece la nueva instancia.

Solución

1. Inicie sesión en la consola de DCS.
2. En la esquina superior derecha, elija **Enterprise** >**Project Management**. En la página mostrada, haga clic en **View Resource** en la fila que contiene el proyecto de empresa deseado.
3. Haga clic en la ficha **Permissions**. A continuación, haga clic en **Assign Permissions** en la ficha **User Groups**.

4. Seleccione los grupos de usuarios a los que desea asignar permisos y haga clic en **Next**.
5. Seleccione **DCS UserAccess** y haga clic en **OK**.

11 Uso de Memcached

11.1 ¿Puedo volcar datos de instancia de DCS Memcached para análisis?

No.

11.2 ¿Qué versión de Memcached es compatible con DCS for Memcached?

DCS for Memcached se basa en Redis 3.0 y es compatible con Memcached 1.5.1.

11.3 ¿Qué estructuras de datos admite DCS for Memcached?

Solo se admite la estructura clave-valor.

11.4 ¿DCS for Memcached apoya el acceso público?

No.

Si el acceso público está deshabilitado para una instancia de DCS, no puede acceder a ella en los entornos locales y solo puede acceder a ella a través de un ECS que está en la misma VPC que la instancia. Las VPC se utilizan para garantizar la seguridad de la red de los servicios de la nube pública.

Durante el desarrollo y la depuración de aplicaciones, puede conectarse a una instancia de DCS desde su entorno local mediante un ECS que puede comunicarse con su instancia para reenviar sus solicitudes. Para obtener más información, consulte **Uso de túnel SSH para el acceso público a una instancia de DCS**.

11.5 ¿Puedo modificar parámetros de configuración de instancias de DCS Memcached?

Configuración de parámetros solo se permite cuando las instancias DCS están en el estado **Running**.

Para obtener más información, consulte la sección **Modificación de parámetros de configuración**.

11.6 ¿Cuáles son las diferencias entre DCS for Memcached y Memcached autohospedado?

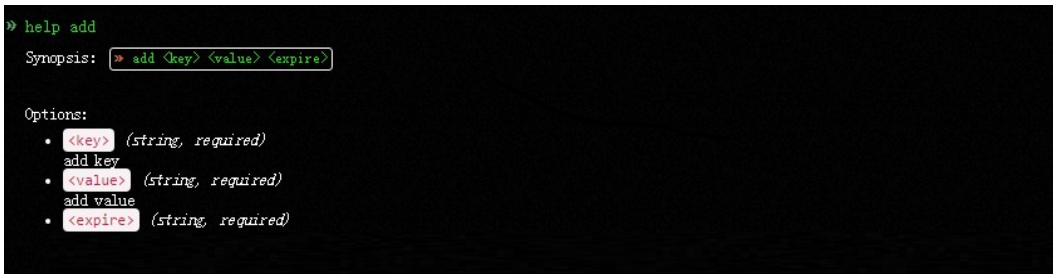
Tabla 11-1 describe las diferencias entre DCS for Memcached y Memcached autohospedado.

Tabla 11-1 Comparación de DCS for Memcached y Memcached autohospedado

Concepto	DCS for Memcached	Memcached autohospedado
Despliegue	Fácil de desplegar. DCS for Memcached se puede utilizar inmediatamente sin necesidad de preocuparse por el hardware o el software.	Implica operaciones y configuraciones complicadas.
Disponibilidad	Las instancias principal/en espera utilizan modo de espera sin interrupción para garantizar los servicios estables. Si el nodo principal es defectuoso, el nodo de caché en espera se convertirá automáticamente en el nodo principal para evitar un único punto de falla.	Requiere las configuraciones adicionales.
Seguridad	Utiliza la VPC y los grupos de seguridad para el control de seguridad de acceso a la red.	Requiere que usted mismo diseñe y despliegue un mecanismo de seguridad.
Escalamiento vertical	Compatible con escalamiento vertical en línea en la consola.	Requiere el hardware adicional y reiniciar el servicio.

11.7 ¿Qué políticas utiliza DCS for Memcached para tratar los datos caducados?

DCS for Memcached le permite establecer el tiempo de caducidad de los datos almacenados en función de los requisitos de servicio. Por ejemplo, puede establecer la hora **expire** al realizar la operación **add**.



De forma predeterminada, los datos no se desalojan de las instancias de DCS Memcached. En la versión actual de DCS for Memcached, puede seleccionar una política de desalojo.

Para obtener detalles sobre los seis tipos de políticas de desalojo de datos, consulte **¿Cuál es la política predeterminada de desalojo de datos?**

11.8 ¿Cómo selecciono las AZ al crear una instancia de DCS Memcached?

Las diferentes AZ dentro de una región no difieren en funciones.

En general, el despliegue de instancia dentro de una AZ presenta una menor latencia de red, mientras que el despliegue cruzado garantiza la recuperación ante desastres. Si su aplicación requiere una latencia de red más baja, elija el despliegue de AZ única.

DCS for Memcached admite el despliegue entre las AZ. Al crear una instancia de DCS Memcached en la consola de DCS, puede seleccionar cualquier AZ en la misma región que ECS para la comunicación entre el ECS y la instancia. Para una menor latencia de red, seleccione la misma AZ que su ECS.

Se supone que tiene un ECS que está en **AZ B** en la región del **CN-Hong Kong**. Al comprar una instancia de DCS Memcached, puede seleccionar cualquier AZ de **CN-Hong Kong**. Si selecciona **AZ B** en **CN-Hong Kong**, su instancia puede comunicarse con su ECS con una latencia de red más baja.

Tenga en cuenta que solo puede haber una AZ disponible debido a la insuficiencia de recursos al crear una instancia de DCS Memcached. Esto no afecta al uso normal de DCS.

