Distributed Cache Service

Data Migration Guide

Issue	01
Date	2024-09-24

Huawei Cloud Computing Technologies Co., Ltd.

Contents

1 Getting Started with Redis and Reading and Writing Data Using Web CLI

This section takes the example of creating a single-node Distributed Cache Service (DCS) Redis 5.0 instance, connecting to it, and performing data read and write operations on the DCS console to get you quickly started with DCS.

There are three steps to get started with DCS:

1. **Preparations**

   Before creating a Redis instance, an authenticated Huawei Cloud account with balance, users with required permissions, and a set of VPC and subnet should be prepared.

2. **Creating a DCS Redis Instance**

   When creating a Redis instance, you can customize it with the required type and specifications, and configure a VPC and a subnet.

3. **Accessing a Redis Instance and Reading and Writing Data**

   Access Redis instances and run Redis commands to read and write data on a client compatible with Redis or using Web CLI on the DCS console.

Preparations

1. **Register a Huawei Cloud account and authenticate it**.

   If you already have one, go to **2**. If you do not have one, do as follows:

   a. Log in to the **Huawei Cloud official website**, and click **Register**.

   b. Complete the registration as prompted. For details, see **Signing up for a HUAWEI ID and Enabling Huawei Cloud Services**.

   Your personal information page is displayed after the registration completes.

   c. Complete real-name authentication for individual or enterprise users by referring to **Real-Name Authentication**.

   📖 **NOTE**

   Real-name authentication is required only when you buy or use resources in the Chinese mainland.

2. **Top up your account**.

   Ensure enough balance in your account.

   – About Redis instance prices, see **Price Calculator**.

   – About topping up an instance, see **Top-Up and Repayment**.

3. **Add DCS permissions**.

   Before creating a Redis instance and its dependencies, specific permissions are required. For details about how to add user permissions, see **Creating a User and Granting DCS Permissions**.

4. **Create a VPC and a subnet.**

   A DCS instance runs in a virtual private cloud (VPC) and has a subnet. Prepare a VPC and a subnet before creating a DCS instance. For more information, see **Creating a VPC**. If you already have an available VPC and subnet, use them.

   📖 NOTE

   ● The VPC must be created in the same region as your DCS Redis instance.

   ● In VPC and subnet creation, retain the default settings unless otherwise specified.

Creating a DCS Redis Instance

Step 1 Go to the **Buy DCS Instance** page.

Step 2 Configure instance specifications. For details, see **Table 1-1**.

Table 1-1 Configuring instance specifications

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	Retain the default setting.
Project	Retain the default setting.
Cache Engine	This example uses **Redis**.
Edition	Select **Basic**.
CPU Architecture	Select **x86**.
Version	Select **5.0** in this operation.
Instance Type	Select **Single-node** in this operation. Single-node Redis instances use single nodes, do not persist data, and cost low. They are applicable in development and testing.
AZ	Each region contains multiple AZs. This example retains the default AZ.
Instance Specification	Select **128 MB** in this operation.
VPC	Select the VPC and subnet in **Preparations**.

Parameter	Description
IP Address	**Automatically-assigned IP address** and **Manually-specified IP address** are available. The port can be customized, or 6379 will be used by default if you leave it blank. This example uses **Automatically-assigned IP address** and the default port 6379.
Security Group	Skip it.
Name	Instance name. By default, a random name is generated. Retain it.
Enterprise Project	An enterprise project manages cloud resources by gathering relevant ones together. The default option **default** is available. If you cannot select a required enterprise project, see **Why Can't I Select the Required Enterprise Project When Creating a DCS Instance?**
Password Protected	Select **Yes** and enter **Password** and **Confirm Password**.
Parameter Configuration	Select **Default templates**.
Quantity	Indicates how many instances you want to buy. One instance will be created by default.
(Optional) More Settings	This example does not use these settings.

Step 3 Check the estimated price in the lower left corner of the page. You can click **Pricing details** to learn more.

Step 4 Click **Next**.

Step 5 Confirm the instance settings, and click **Submit**.

The instance has been created successfully if it is in the **Running** state.

Step 6 After the task is successfully submitted, the **Cache Manager** page is displayed. When the new instance is in the **Running** state, the instance is created successfully.

----End

## Accessing a Redis Instance and Reading and Writing Data

This section takes the example of accessing a DCS Redis instance by using Web CLI on the console.

📖 **NOTE**

To access a DCS Redis instance on a compatible client, see **Accessing Redis on a Client**.

Step 1 In the navigation pane, choose **Cache Manager**. In the **Operation** column of the instance, choose **More** > **Connect to Redis**, as shown in the following figure.

Figure 1-1 Accessing Web CLI
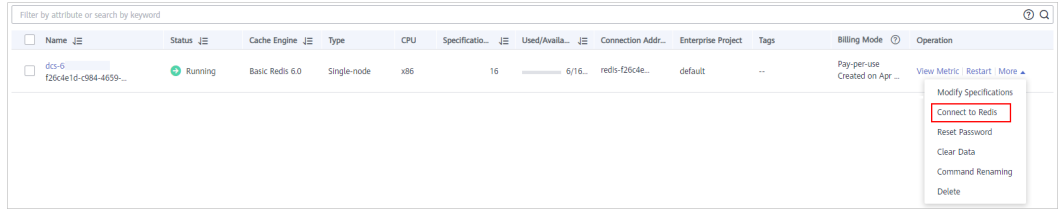
Step 2 Enter the password set in instance creation and go to Web CLI.

> 📖 **NOTE**
>
> - If no operation is performed for more than 5 minutes, the connection times out. You need to enter the password to access the instance again.
> - You do not need to enter a password for accessing a password-free DCS Redis instance.

Step 3 Single-node instances support multiple databases (multi-DB). Select one of the databases. Select "Database 0" here.

Figure 1-2 Selecting a Database
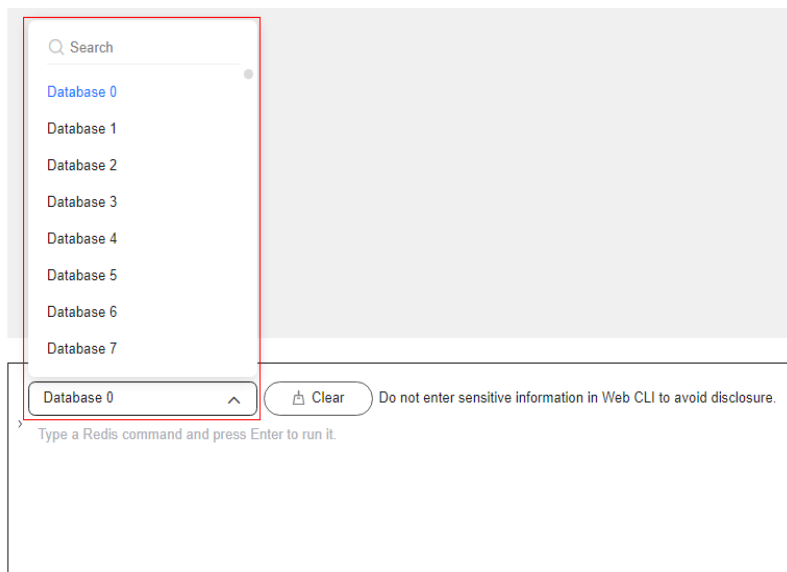
Step 4 Enter Redis commands in the input box. For example, run the **SET** command to write a data name **KEY_NAME** and data value **VALUE**, and press **Enter**. The data is written when "OK" is returned. Write another **ABC** data of **123** again.

```
> SET KEY_NAME VALUE
 OK
> SET ABC 123
 OK
```

Step 5 Run the **GET** command to read the written data.

```
> GET KEY_NAME
 VALUE
> GET ABC
 123
```

----End

Related Information

- For details about related concepts of DCS, see **Basic Concepts**.
- For details about the supported commands in DCS, see **Command Compatibility**.
- For details about DCS monitoring metrics, see **DCS Metrics** and **Configuring Alarm Rules for Critical Metrics**.
- For suggestions on using DCS instances, see **Suggestions on Using DCS**.

2 Common Practices

This section describes common practices of DCS to help you better use it.

Practice	Description
Serializing Access to Frequently Accessed Resources	In Internet scenarios such as seckill, the system needs multiple machines to run concurrently to handle traffic burst. If two users' requests arrive at the same time but on two different machines. Although the two requests can be handled spontaneously, an inventory oversold or disordered access problem may still occur. This is because a machine has a unique lock and a machine's lock only takes effect on threads in the Java VM where the machine runs. To serialize access to resources, you can use DCS Redis instances for distributed locking.
Merging Game Servers with DCS	Merging game servers takes place when large-scale online games start a new server (zone) or merge old and new servers. During this process, game developers must consider how to synchronize data among different servers. With the pub/sub message queuing mechanism of DCS for Redis, data changes can be published to Redis channels. Other game servers can subscribe to the channels to receive messages of changes. This practice describes how to synchronize servers using Redis.
Connecting a Redis Client to DCS Through CCE	With the development of the container technology, more and more applications are deployed in containers. This practice describes how to deploy a Redis client in a Cloud Container Engine (CCE) cluster container and connect it to DCS.