

Blockchain Service

Getting Started

Issue 01
Date 2024-06-17



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

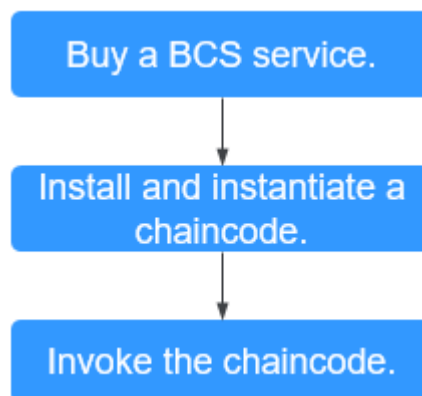
1 Process of Using BCS.....	1
2 Purchasing a BCS Instance.....	2
3 Installing and Instantiating a Chaincode.....	3
4 Invoking Chaincodes.....	7

1 Process of Using BCS

Huawei Cloud Blockchain Service (BCS) is a blockchain technology platform for enterprises and developers. BCS helps you quickly deploy, manage, and maintain blockchain networks on Huawei Cloud. BCS reduces the threshold for using blockchain and allows you to focus on the development and innovation of your own business.

Figure 1-1 shows the process of using BCS.

Figure 1-1 Process of using BCS



Process description:

1. **Buy** a BCS instance.
Register a Huawei Cloud account and purchase a BCS instance.
2. **Install and instantiate a chaincode.**
After purchasing a BCS instance, install and instantiate a chaincode.
3. **Invoke the chaincode.**
Download an SDK to the service application and invoke the chaincode by using the SDK.

2 Purchasing a BCS Instance

Prerequisites

Before using BCS, you must have a Huawei Cloud account. For details, see [Account Registration Process](#).

Procedure

- Step 1** Go to the page for purchasing [enhanced Hyperledger Fabric instances](#).
- Step 2** On the page that is displayed, specify the enterprise project, enter a BCS instance name (**demo**), and set the initial password for resource access. Retain the default settings for other parameters. For details, see [Deployment Using a CCE Cluster](#).

NOTE

You are advised to set the BCS instance name to **demo**. Then, the name of the .yaml file in the downloaded SDK configuration file will be **demo-channel-sdk-config.yaml**.

If you use another name, for example, **bcs123**, the name of the .yaml file in the downloaded SDK configuration file will be **bcs123-channel-sdk-config.yaml**. When configuring an application, change the file name to **demo-channel-sdk-config.yaml**. For details, see [Invoking Chaincodes](#).

- Step 3** Configure the parameters as prompted and finish the BCS instance purchasing process.

Wait for several minutes. After a message is displayed indicating successful creation, check the status of the instance and organizations. If they are **Normal**, the BCS instance deployment is completed.

----End

3 Installing and Instantiating a Chaincode

To obtain the chaincode file used in the example, go to the BCS console and click **Use Cases**. Download the example Go chaincode in the **Getting Started** area.

Installing a Chaincode

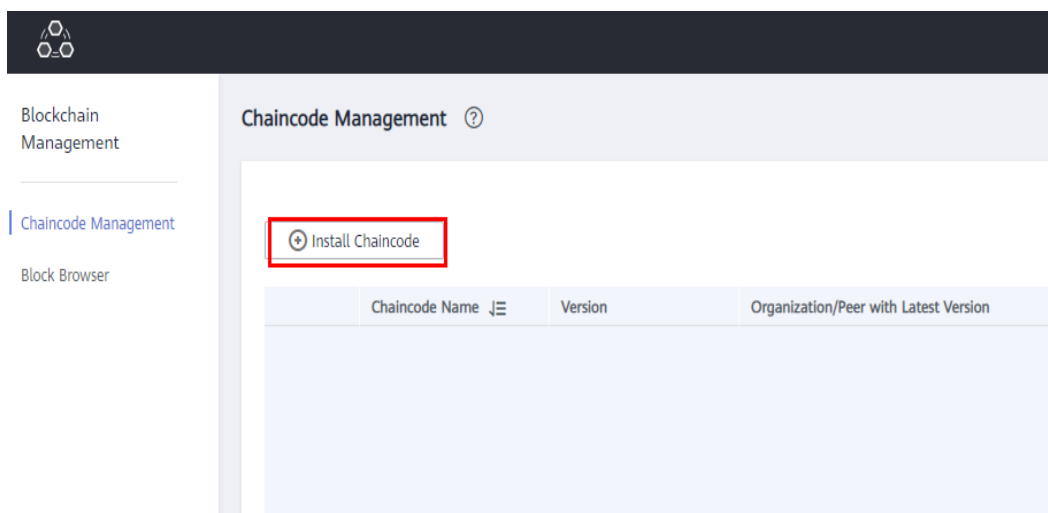
Step 1 Click **Manage Blockchain** on the card of the BCS instance you just created.

Step 2 On the login page, enter the username and password, and click **Log In**.

NOTE

The username is **admin**, and the password is the **Blockchain Mgmt. Initial Password** set when you created the BCS instance. If you have not set this password, use the resource access initial password. For details, see [Purchasing a BCS Instance](#). For security purposes, change the password periodically.

Step 3 Click **Install Chaincode** in the upper left corner of the page.



Step 4 Enter the chaincode name and version number, select the peers where the chaincode is to be installed, specify the programming language of the chaincode, and add the chaincode file.

Table 3-1 Chaincode installation configurations

Parameter	Description
Chaincode Name	Enter chaincodedemo
Chaincode Version	Enter 1.0
Ledger Storage	Default option: File database (goleveldb) .
Peer	Manually select organizations and peers.
Language	Select Golang .
Chaincode File	Add a chaincode file. To obtain the chaincode file used in the example, go to the BCS console and click Use Cases . Download the example Go chaincode in the Getting Started area.
Chaincode Description	Enter a description of the chaincode.
Code Security Check	This option is displayed only when the chaincode language is Golang. Enable this option to check code security.

Figure 3-1 Installing a Chaincode

Install Chaincode

* Chaincode Name

* Chaincode Version

Ledger Storage File database (goleveldb)

Select All Peers

Organization & Peer

Language

Chaincode File example01.zip

Chaincode Description
0/500

Code Security Check

Step 5 Click **Install**.

Step 6 Click next to a chaincode name to view the details.

Step 7 Click **Download** in the **Operation** column to view the check result. (The following example is for reference only.)

NOTE

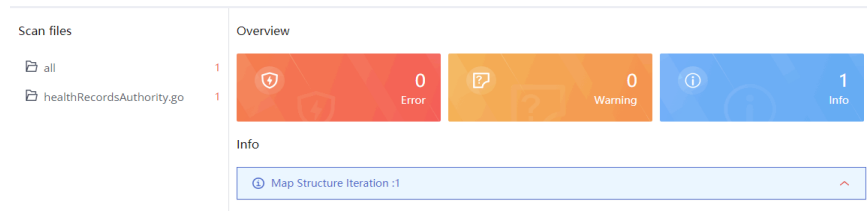
If **Code Security Check** is not enabled, no check report will be generated, and the **Download** button will not be displayed.

Figure 3-2 Downloading the check report

Chaincode Version	SHA-256 Hash	Description	Installed	Operation
1.0	3455d0016a31349749ff57c7b00c131782e130832a0a1afe61f1...		Nov 15, 2021 10:26:23 GMT+08:00	Download Delete

- Decompress the package and open the HTML file to view the check result details. There are three types of issues: error, warning, and info. Error-level issues must be resolved. Otherwise, the chaincode functions will be affected. Warning-level issues can be handled by reconstructing the code. Info-level issues can be handled selectively as required.

Figure 3-3 Scanned files



- For example, there is an info-level issue in the preceding figure. You can click the issue to view its details, including a brief description, wrong example, scanning details, modification advice, and revision example.

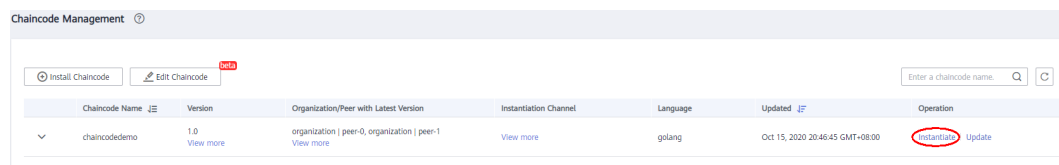
NOTE

Modify the code based on the chaincode check result and update the chaincode or install it again.

----End

Instantiating a Chaincode

Step 1 After installing the chaincode, click **Instantiate** in the **Operation** column that contains a target chaincode.



Step 2 Specify the channel, chaincode version, initialization function, endorsement policy, endorsing organizations, and other parameters.

Instantiate Chaincode ?

Chaincode Name	bbbbbbbb
Channel	<input type="text" value="channel"/>
Chaincode Version	<input type="text" value="1.0"/>
* Initialization Function	init Chaincode function that will be invoked
Chaincode Parameters	<input type="text" value="For example, a,200,b,250"/> Enter the parameters of the initialization function init(). Separate multiple parameters with commas.
Endorsement Policy	<input checked="" type="radio"/> Endorsement from any of the following organizations <input type="radio"/> Endorsements from all of the following organizations
Endorsing Organizations	<input type="text" value="organization"/>
Privacy Protection Configuration ?	<input checked="" type="radio"/> No <input type="radio"/> Yes

Step 3 Click **Instantiate**. Wait 2 to 3 minutes for the instantiation on the specified channel to complete.

----End

4 Invoking Chaincodes

Prerequisites

- JDK has been installed and the Java development environment has been configured. The JDK version must be 1.8 (64-bit). If you have installed JDK, run the **java -version** command to check the JDK version.
- Log in to the BCS console and click **Use Cases**. Download the example Java chaincode in the **Getting Started** area. In Windows, download and decompress the package to **C:/jvasdkdemo**. In Linux, download and decompress the package to **/root/jvasdkdemo**.

Configuring the Application

- Step 1** On the **Instance Management** page, click **Download Client Configuration** on an instance card.
- Step 2** Select **SDK Configuration File**, and enter the chaincode name and certificate path. Retain the default values for other parameters.
 - The chaincode name must be the same as the name specified during chaincode installation and instantiation, for example, **chaincodedemo**.
 - The certificate path is the final path for storing the certificate for application compilation. If the certificate path changes, you must manually change all certificate paths in the SDK configuration file.
Windows: **C:/jvasdkdemo/config**
Linux: **/root/jvasdkdemo/config**
- Step 3** Select **Orderer Certificate**.
- Step 4** Select **Peer Certificates**, select **organization** for **Peer Organization**, and select **Administrator certificate**.
- Step 5** Click **Download** to download the SDK configuration file and the administrator certificates for the **demo-orderer** and **organization** organizations.

NOTE

- The BCS instance name is **demo**, so the name of the .yaml file in the downloaded SDK configuration file is **demo-channel-sdk-config.yaml**.
- If you use another name, for example, **bc123**, the name of the .yaml file in the downloaded SDK configuration file will be **bc123-channel-sdk-config.yaml**. You must manually change the file name to **demo-channel-sdk-config.yaml**.

Step 6 Decompress **demo-config.zip** and copy the **orderer** and **peer** folders and the **sdk-config.json** and **sdk-config.yaml** files to the **config** directory where the demo is stored (**C:/javasdkdemo/config** in Windows and **/root/javasdkdemo/config** in Linux).

NOTE

The **demo-channel-sdk-config.yaml** file must exist in the **config** directory of the **javasdkdemo** project.

----End

Deploying the Application

Step 1 Go to the directory where the demo is stored (**C:/javasdkdemo** in Windows and **/root/javasdkdemo** in Linux) and run the following command:

```
java -jar javasdkdemo.jar
```

Each time the command is successfully executed, the key-value pair **<testuser,100>** is saved to the blockchain. If you query key **testuser**, the value is **100**.

The command output is as follows:

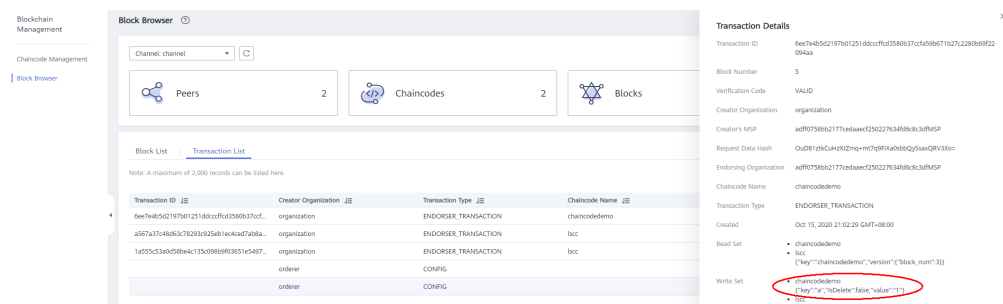
Figure 4-1 Execution result

```

$ java -jar javasdkdemo.jar
2020-10-12 17:57:41.141 WARN [org.hyperledger.fabric.sdk.helper.Config] - Failed to load any configuration from: config.properties. Using toolkit defaults
2020-10-12 17:57:43.081 INFO [org.hyperledger.fabric.sdk.Channel] - Channel ChannelId: 1, name: channel) eventThread started shutdown: false thread: null
2020-10-12 17:57:45.253 INFO [handler.Main] - [insert new data <testuser, 100> success] Store key=testuser to the blockchain
2020-10-12 17:57:45.273 INFO [handler.Main] - [query key <testuser> value is 100] Retrieve the value of the testuser key from the blockchain
    
```

Step 2 View the results in the **Block Browser**.

Figure 4-2 Viewing the results



----End