Blockchain Service

FAQs

Issue 01

Date 2025-11-28

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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 Enhanced Hyperledger Fabric

1.1 Billing

1.1.1 FAQs Related to BCS Billing

1. How do I change the billing mode of a BCS instance from pay-per-use to yearly/monthly?

   Currently, the billing mode of a BCS instance cannot be changed from pay-per-use to yearly/monthly.

2. How do I unsubscribe from a yearly/monthly BCS instance?

   Choose **More** > **Unsubscribe** on the instance card. After the unsubscription application is approved, the remaining fees paid for the instance will be refunded.

3. What can I do if my BCS instance is abnormal because my account is in arrears?

   Renew your BCS instance at **Billing Center** > **Renewal**, and the instance status will become normal within 5 to 10 minutes. If the BCS instance remains abnormal, refer to **Abnormal Instance Statuses**.

1.2 Instance Management

1.2.1 Consultation

1.2.1.1 How Do I Determine Whether a Blockchain Is Necessary?

To determine whether the blockchain technology is suitable for your project, answer the following questions in sequence:

- Need multiple parties share data?

  Will all business participants benefit from a complete and reliable record sharing system?

- Need multiple parties update data?

  Will data accuracy and update timeliness be enhanced if multiple participants can record and propagate concurrent transactions?

- Must data be verified?

  Can tamper-proof transactions in an untrusted environment improve the transaction throughput and reliability of partners?

- Can a central institution be removed?

  Is removal of a central institution helpful to reduce costs and transaction complexity?

If your answer is "Yes" for all of the above questions, then your project needs the blockchain technology.

1.2.1.2 What Underlying Framework Is Used for Huawei Cloud BCS?

Huawei Cloud BCS uses the Hyperledger open-source framework.

Hyperledger is a blockchain open-source project hosted by the Linux Foundation to establish an underlying architecture for the distributed ledger platform oriented to multiple application scenarios. Hyperledger has sub-projects including Hyperledger Fabric, the most important one, and many other related projects derived on top of it. Developers from various sectors, such as finance, banking, IoT, supply chain, and manufacturing, have contributed towards this project seeking to build cross-field blockchain applications.

Based on the Hyperledger framework, Huawei Cloud BCS offers a highly available and secure blockchain platform boasting a superb performance.

1.2.1.3 Can BCS Instances Deployed on the Public Cloud Access Blockchain Nodes on Other Clouds?

BCS instances now can be deployed only on the public cloud. Each user can deploy multiple BCS instances. Multiple public cloud users' BCS instances can form a consortium blockchain.

Huawei Cloud-dominated multi-cloud hybrid deployment of BCS instances has been planned, and the launch time will be determined soon.

Blockchain applications of other vendors cannot connect to the BCS.

1.2.1.4 What Competitive Advantages Does Huawei Cloud BCS Have?

BCS provides multiple options of consensus algorithms, visualized smart contract (chaincode) management, and security and privacy protection (using OSCCA-published cryptographic algorithms, homomorphic encryption, and zero-knowledge proofs).

1.2.1.5 In Which Direction and What Capabilities Will Huawei Cloud BCS Develop?

BCS will improve its competitiveness in terms of availability, security, performance, and blockchain ecosystem (such as the smart contract library and blockchain tool

kits). It will continuously develop to provide underlying technology services for enterprise-grade commercial blockchains.

1.2.1.6 What Are the Specifications of VMs to Be Purchased for BCS?

Purchase virtual machine (VM) resources to deploy and run the BCS instances. VMs of the following specifications are recommended.

Table 1-1 Suggestions on VM purchase

Business Phase	Consensus Algorithm	Recommended VM Specifications (Minimum)
POC	Fast Byzantine Fault Tolerance (FBFT)	1 VM with 8 vCPUs and 16 GB memory
Commercial use	-	It is recommended that each peer uses 4 vCPUs and 8 GB memory. Determine the specifications based on the service scale and number of users. You can contact the BCS technical support for help.

1.2.1.7 How Do I Get Access to the Partners of Huawei Cloud BCS for More Services?

If you have requirements for service chaincode or client app design and development, reach us by **sales@huaweicloud.com**. BCS and its partners will provide you with comprehensive solutions based on your business and Huawei Cloud advantages and features.

1.2.1.8 What Are the Differences Between Channel Isolation and Privacy Protection?

Channel isolation: A channel isolates the ledger data of a transaction from other transaction data in a consortium blockchain to ensure confidentiality. Each channel can be considered a sub-blockchain and corresponds to a specific ledger. The ledger in a channel is invisible to other channels.

Privacy protection: Privacy is ensured within each channel because different members in a channel can have different access permissions. For example, member A has the permissions to access certain data, but member B, who does not have relevant permissions, cannot access the specified data.

In short, privacy protection isolates data for a member from other members in same channel, while channel isolation isolates data for all members in a channel from other channels.

1.2.1.9 How Well Does BCS Perform?

The following performance data is obtained from the pressure test carried out by using a 32 vCPUs | 64 GB ECS and two clients.

Table 1-2 Performance in different scenarios

Scenario	Performance
ECDSA + FBFT	Supported concurrency: 50; TPS (consistency maintained): 6504
OSCCA-published cryptographic algorithms + FBFT	Supported concurrency: 50; TPS (consistency maintained): 5698

1.2.1.10 Does BCS Support Customized Development?

No. BCS does not support customized development, but it provides demos. For details, see the **Developer Guide**.

1.2.1.11 When Do I Need to Hibernate or Wake an Instance?

Scenario

You can hibernate unnecessary pay-per-use BCS instances and wake them later. Instances in hibernation are unavailable.

Benefits

BCS instances in hibernation do not incur any management fees. This enables accurate pay-per-use billing and helps enterprises reduce unnecessary investments.

NOTE

After an instance is hibernated, the resources configured for it, such as Elastic Cloud Server (ECS) nodes and Elastic Volume Service (EVS) disks, are billed according to their billing modes.

1.2.2 Service Usage

1.2.2.1 How Do I Check Whether the ICAgent Is Installed for the Cluster?

ICAgent is a log collector. It runs on each host to collect metrics, logs, and application performance data in real time.

If ICAgent is not installed for the cluster used by a BCS instance, the log data aging and O&M data collection functions may become unavailable, the root directory space may be exhausted, and the instance may be interrupted.

Perform the following operations to check the ICAgent status. If the status is **Uninstall**, install the ICAgent.

Procedure

Step 1 Log in to the BCS console, on the **Instance Management** page, click the **Enhanced Hyperledger Fabric** tab page.

Step 2 Click an instance name. On the **Basic Information** page, click **More** in the upper right corner.

Step 3 Log in to the Application Operations Management (AOM) console, choose **Configuration Management** > **Agent Management**, select the cluster in the upper right corner, and check the ICAgent status.

- If the ICAgent status is **Running**, the ICAgent has been installed and is running properly.
- If the ICAgent status is **Uninstalled**, refer to **Installing the ICAgent** to install the ICAgent.

----End

1.2.2.2 What Can I Do If I Can't Open the Blockchain Management Console?

Symptom

The Blockchain Management console cannot be opened.

Solution

1. Check whether you are using Internet Explorer to log in to the Blockchain Management console.

   If you use Internet Explorer, you may fail to access the Blockchain Management console and see a message indicating that the certificate is untrusted. In this case, check **the Internet Explorer instructions** to resolve the problem.

2. Check whether the BCS instance status is abnormal.

   If the BCS instance is in the **Abnormal**, **EIP abnormal**, **Frozen**, or **Unknown** state, the Blockchain Management console will become unavailable. For details, see **Abnormal Instance Statuses**.

1.2.2.3 What Should I Do If My BCS Instance Remains in the Creating State?

The possible cause is that the disk fails to be mounted.

Solution

1. Log in to the node in the Cloud Container Engine (CCE) cluster where the BCS instance is deployed, and run the following command to check the DNS address in the POD zone. If the DNS address in the POD zone is incorrectly configured, the domain name cannot be resolved and the disk fails to be mounted.

   ```
   vi /etc/resolve.conf
   ```

2. If the fault persists, contact technical support.

1.2.2.4 What Should I Do If a Peer Restarts Frequently with the Error Message "PanicDB not exist"?

1. Go to the **/home/paas/evs/baas/{Service ID}/{Container ID}/** directory of the peer container and delete the **production** folder.

2. Restart the peer and agent containers, obtain the ledger again, and add the peer to the channel.

1.2.2.5 What Can I Do If the CPU Usage of a Blockchain Node Reaches 100%?

The user node may have been attacked by viruses. If this happens, perform the following operations:

- Use strong passwords that meet the requirements for all accounts (including system accounts and application accounts).
- Use security groups to control access over specific ports. For special service ports, use fixed source IP addresses, VPNs, or bastion hosts to establish your own O&M channel.
- Periodically back up data (VM internal backup, remote backup, and backup on and off the cloud) to protect data against encrypting ransomware attacks.

1.2.2.6 Why Can't I Log In to the Blockchain Management Console?

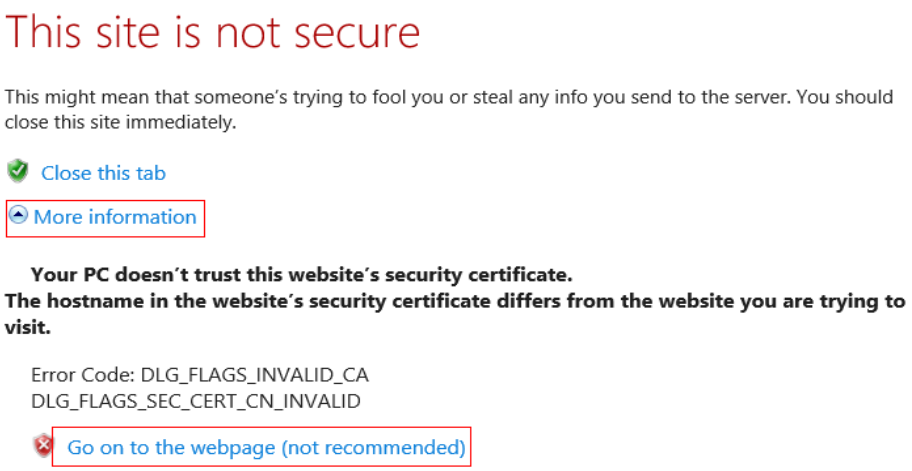
You may need to perform extra steps in your browser before you can be redirected to the Blockchain Management console.

- Internet Explorer

  NOTE

  These instructions are for reference only. The actual browser pages may vary depending on browser versions, but the operations are similar.

  a. Open Internet Explorer and enter the address of the Blockchain Management console in the address box.

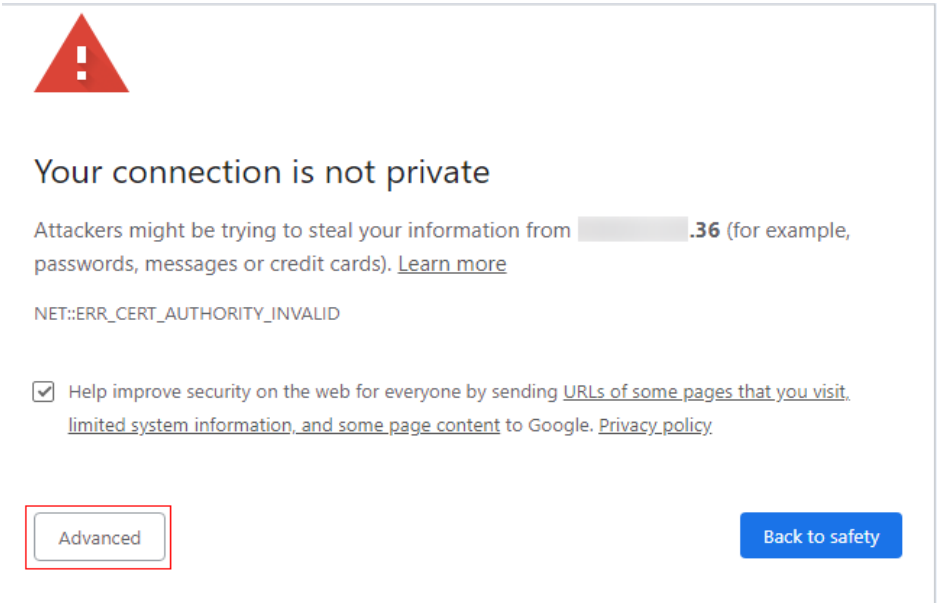
  b. Click **More information** > **Go on to the webpage** and the login page will be displayed.

- Google Chrome

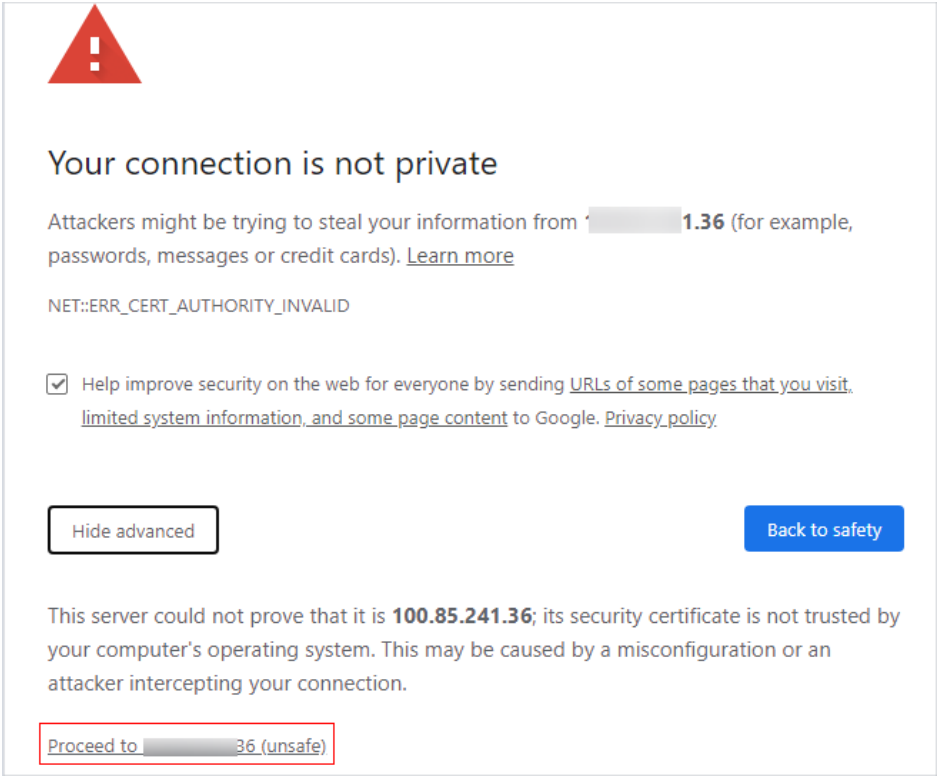
  NOTE

  These instructions are for reference only. The actual browser pages may vary depending on browser versions, but the operations are similar.

a. Open Google Chrome and enter the address of the Blockchain Management console in the address box.



b. Click **Advanced**.



c. Proceed to the login page.

1.2.2.7 BCS.4009100: System Error

Symptom

A system error message is displayed on the **Instance Management** page.

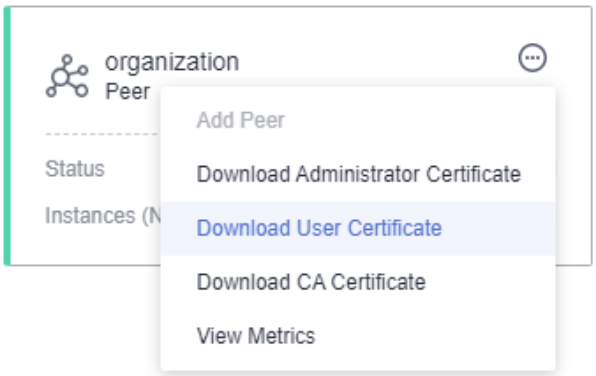
Solutions for Common Scenarios

- **Scenario 1**: The error occurs when you click any button on the **Instance Management** page.
  a. Possible cause 1: The RegionLB component is abnormal.
     Solution: Check the buttons on the Application Operations Management (AOM), CCE, and ServiceStage consoles. If similar errors occur, contact the RegionLB technical support.
  b. Possible cause 2: The BCS backend is abnormal and does not respond to requests.
     Solution: Press F12, click the **Network** tab, and check the request for which an error is reported. Provide the headers and preview information to the BCS technical support.
- **Scenario 2**: The error occurs only when you click certain buttons on the **Instance Management** page.
  Solution: Press F12, click the **Network** tab, and check the request for which an error is reported. Provide the headers and preview information to the BCS technical support.

1.2.2.8 How Can I Obtain Private Keys and Certificates for Enhanced Hyperledger Fabric Blockchains?

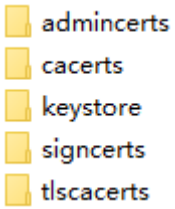
Download the private keys and certificates on the BCS console or generate them using OpenSSL.

- To obtain the private key and certificate of a single user, download them on the BCS console.
  a. Log in to the BCS console.
  b. In the navigation pane on the left, click **Instance Management**. Click an instance to check its details.
  c. In the **Blockchain Organizations** area, click to download the user certificate.

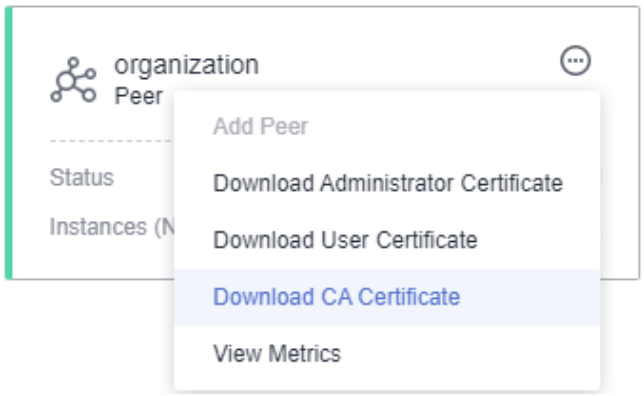


  d. Decompress the downloaded user certificate. The **msp** folder contains the user private key (**keystore**) and certificate (**signcerts**), as shown in the following figure.

Figure 1-1 File directory

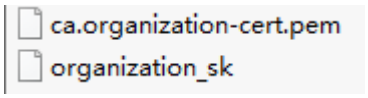


- To generate the private keys and certificates of multiple users, use OpenSSL.
  a. Download the CA certificates and private keys.
     i. Log in to the BCS console.
     ii. In the navigation pane on the left, click **Instance Management**. Click an instance to check its details.
     iii. In the **Blockchain Organizations** area, click to download the CA certificate.



     iv. Decompress the downloaded CA certificate to obtain the following files:

Figure 1-2 Decompressed files



  b. Generate a new ECC private key.
     i. Generate a private key with prime256v1.
     ```
     openssl ecparam -name prime256v1 -genkey -out user-key_.pem
     ```
     ii. Convert the key format to PKCS#8.
     ```
     openssl pkcs8 -topk8 -nocrypt -in user-key_.pem -out user-key
     ```
  c. Generate a certificate request file.
  ```
  openssl req -new -key user-key -out user-csr.pem
  ```
  d. CA issues the certificate.
  ```
  openssl x509 -req -in user-csr.pem -out user-cert.pem -CA ca.organization-cert.pem -CAkey organization_sk -CAcreateserial -days 3650
  ```
  e. A CA-signed certificate file is **user-cert.pem**, and the corresponding private key is **user-key**.

1.2.2.9 Why Does Chaincode Instantiation Fail When I Deploy a Fabric v1.4 Instance Using a v1.19 CCE Cluster?

Currently, Fabric v1.4 instances can be deployed only in v1.15 or earlier CCE clusters. If you want to use v1.19 CCE clusters, set the BCS instance version to Fabric v2.2.

1.2.2.10 Can All Blocks Be Saved As More and More Blocks Are Created?

The increasing number of transactions will lead to blockchain growth and require larger storage space. To ensure all the data is stored, you can:

- Expand the storage space.
  a. Log in to the BCS console and click a BCS instance.
  b. On the BCS instance details page, click **More** on the **Basic Information** tab page and then click **View Details** next to **Network Storage** to obtain **PVC Name**.
  c. Log in to the CCE console, click **Clusters**, and select a target cluster. On the cluster details page, click **Storage**.
  d. On the **PersistentVolumeClaims (PVCs)** tab page, choose **More** > **Scale-out** in the **Operation** column containing the recorded PVC.
- Contact technical support to back up data.

1.2.3 What Can I Do If I Fail to Purchase a BCS Instance?

1.2.3.1 General Checks

Symptom

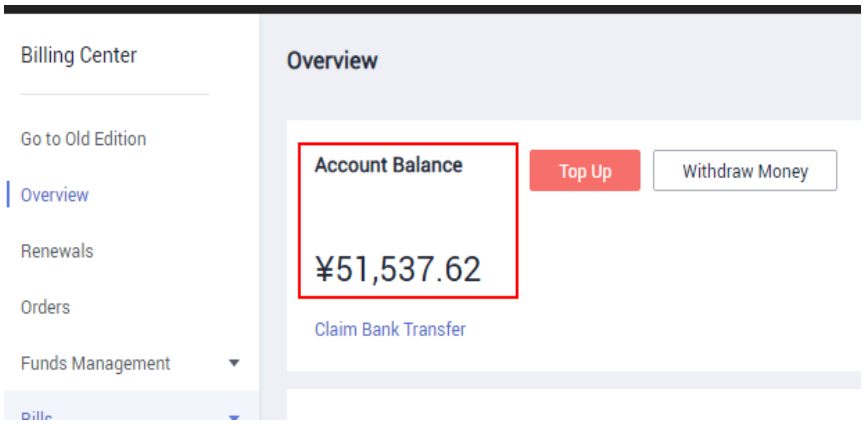
A BCS instance fails to be purchased.

Fault Locating

- **Check item 1: Check whether the account is in arrears.**
- **Check item 2: Check whether the permissions of the IAM user are insufficient.**
- **Check item 3: Check the details error information.**

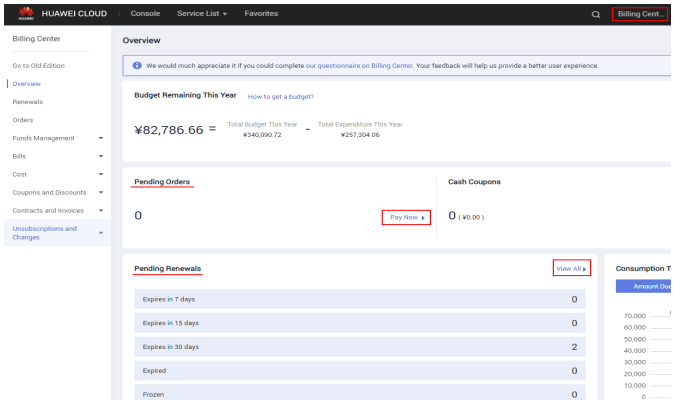
Solution

- Check item 1: Check whether the account is in arrears.
  a. Log in to the BCS console and click **Billing Center** in the upper part of the page.



If **Account Balance** is zero or less than the price displayed when you purchase the BCS instance, the purchase will fail.

b. In the **Pending Orders** area, click **Pay Now**. On the **Orders** page that is displayed, select the order to be paid and complete the payment.



If the payment fails and a message is displayed indicating that the balance is insufficient, the purchase will also fail.

c. In the **Pending Renewal** area, click **View All**. On the displayed page, select the resources to be renewed and click **Renew**.

- Check item 2: Check whether the permissions of the IAM user are insufficient.

  If the instance is purchased by an IAM user, obtain permissions by following instructions in **Permissions Management**.

- Check item 3: Check the details error information.

  a. Log in to the BCS console. In the navigation pane, click **Instance Management**.

  b. Click **Task Details** on the right.

  c. Click **View Details** in the **Operation** column.

  d. In the displayed window, click **View Cause** to analyze the failure. For details, see **Table 1-3**.

Table 1-3 Errors and solutions

Error	Solution
Insufficient cluster quota	CCE Cluster Quota Used Up

Error	Solution
CreateContainerCluster] Create Cluster fail	**Failed to Create a Cluster**
CreateStorageInstance]Create PVC fail,timeout	**Failed to Create a PVC**
CreateBlockchainService]cluster is already used	**Cluster Already In Use**
CreateStorageInstance] PostReqNoCert: response status 400 not OK, resp is: {"errCode":"SFS.TURBO.0010","errMsg":"instance quota exceeds"}	**SFS Turbo File System Quota Exceeded**
not find down eip, need buy one	**No EIP Bound**
CreateContainerCluster]visitInterface Fail,/api/v2/projects/xxx/clusters response state not OK,code:500	**CCE Is Abnormal**
CreateBlockchainSer-vice]DeleteKubeResource failed, clusterID:xxx, resourceName: xxx, err:DELETE:response status 400 not OK, resp is:Cluster Status Not Ready	**Cluster Status Is Abnormal**
CreateContainerCluster]Invalid subnet id xxx	**Subnet Unavailable**
CreateBlockchainService]one OneStep is processing under this domain(xxx),please wait it finished and do it again	**Quick Deployment in Progress**
CreateContainerCluster]cce status check timeout when checkClusterJobStatus...	**CCE Status Check Times Out**
CreateContainerCluster] VisitInterface Fail, PostReqNoCert: response status 400 not OK, resp is: {"message":"\u003cCCE_CM.4000311\u003e No available master flavors | The amount of available master flavors are not enough to create a cluster, in specified azs, need 1, has 0","reason":"BadRequest","code":400}	**Insufficient Master Nodes in the AZ of the CCE Cluster**

1.2.3.2 Detailed Checks

1.2.3.2.1 CCE Cluster Quota Used Up

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

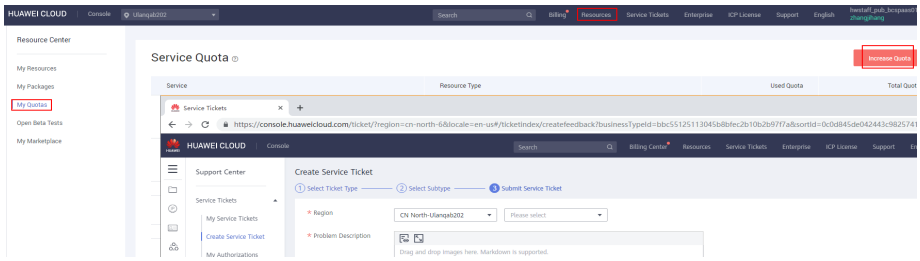
Symptom

The BCS instance fails to be created because the CCE cluster quota has been used up.

Solution

Apply for a higher cluster quota and purchase a BCS instance again.

Log in to the CCE console. In the upper right corner of the page, choose **Resources** > **My Quotas**. On the displayed page, click **Increase Quota**. On the **Create Service Ticket** page, submit a service ticket.



1.2.3.2.2 Failed to Create a Cluster

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The CCE cluster fails to be created during the BCS instance creation.

Solution

Create a cluster first and select the created cluster when buying a BCS instance.

Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

1.2.3.2.3 Failed to Create a PVC

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The PVC fails to be created during the BCS instance creation.

Solution

Step 1 Create a Scalable File Service (SFS) file system.

Log in to the SFS console and click **Create File System** or **Buy Resource Package**..

Step 2 Buy a CCE cluster.

Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

Step 3 Import the SFS file system to the cluster.

In the navigation pane of the CCE console, choose **Resource Management** > **Storage** > **SFS**. Click **Import**, select the target cluster and the project, select the target PVC, and click **OK**.

Step 4 When purchasing the BCS instance, select an existing CCE cluster.

----End

1.2.3.2.4 Cluster Already In Use

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because the CCE cluster has already been used.

Solution

Solution 1: Select an unused CCE cluster when creating a BCS instance.

Solution 2: Create a cluster first and select the created cluster when buying a BCS instance.

To create a cluster:

Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

1.2.3.2.5 SFS Turbo File System Quota Exceeded

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because the SFS Turbo quota limit is exceeded.

Solution

Apply for a higher SFS Turbo file system quota and purchase a BCS instance again.

Log in to the SFS console. In the upper right corner, choose **Resources** > **My Quota** > **Increase Quota**. On the **Create Service Ticket** page, submit a service ticket.

1.2.3.2.6 No EIP Bound

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because no EIP is bound.

Solution

Buy a BCS instance again.

1.2.3.2.7 CCE Is Abnormal

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because CCE is abnormal.

Solution

Deploy your BCS instance using an existing cluster. You can create a CCE cluster first.

Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

1.2.3.2.8 Cluster Status Is Abnormal

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because the cluster status is abnormal.

Solution

- Solution 1: Check whether the cluster is running properly.

  On the CCE console, check the cluster status on the **Clusters** page.

  If the cluster status is normal, purchase a BCS instance again. If the cluster status is abnormal, submit a service ticket for consultation.
- Solution 2: Select an unused CCE cluster when creating a BCS instance.

1.2.3.2.9 Subnet Unavailable

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because the subnet is unavailable.

Solution

Deploy your BCS instance using an existing cluster. You can create a cluster first.

To create a cluster:

Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

1.2.3.2.10 Quick Deployment in Progress

Refund Description

If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because another instance is being created through one-click deployment under the account.

Solution

Buy a new BCS instance after the current quick deployment completes.

1.2.3.2.11 CCE Status Check Times Out

Refund Description

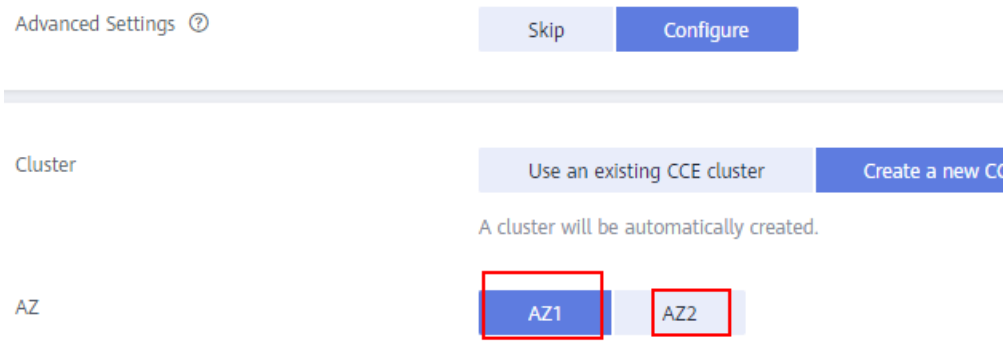
If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because the CCE cluster status check times out.

Solution

- Solution 1: Specify a different availability zone (AZ) when re-creating a BCS instance.



  If the BCS instance fails to be created again, submit a service ticket.

- Solution 2: Create a cluster first and select the created cluster when buying a BCS instance.

  To create a cluster:

  Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

1.2.3.2.12 Insufficient Master Nodes in the AZ of the CCE Cluster

Refund Description

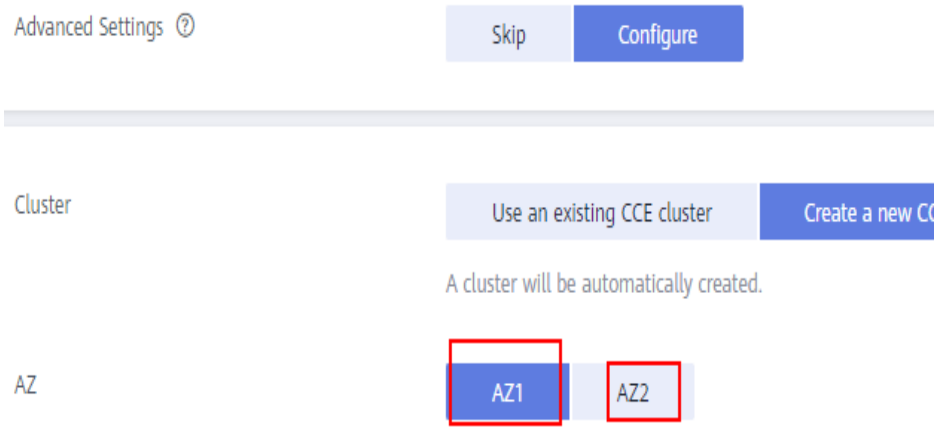
If a BCS instance fails to be created, the deducted fee will be returned to the account within 0.5 to 1 hour.

Symptom

The BCS instance fails to be created because the master node resources are insufficient in the AZ of the CCE cluster.

Solution

- Solution 1: Specify a different AZ when re-creating a BCS instance.



If the BCS instance fails to be created again, submit a service ticket.

- Solution 2: Create a cluster first and select the created cluster when buying a BCS instance.

  To create a cluster:

  Log in to the CCE console and click **Buy Cluster**. For details, see **Buying a CCE Standard/Turbo Cluster**.

1.2.4 Abnormal Instance Statuses

1.2.4.1 What Can I Do If a BCS Instance Is in the Abnormal State?

Symptom

The BCS instance is in the **Abnormal** state.

Fault Locating

Check item 1: Check whether the cluster, storage, and server resources on which the blockchain depends are normal.

Check item 2: Check whether the ECS specifications can meet the requirements.

Solution

- Check item 1: Check whether the cluster, storage, and server resources on which the blockchain depends are normal.

  a. Check the CCE cluster status.

  i. Log in to the CCE console, click **Clusters**, and check the status of the abnormal blockchain's cluster.

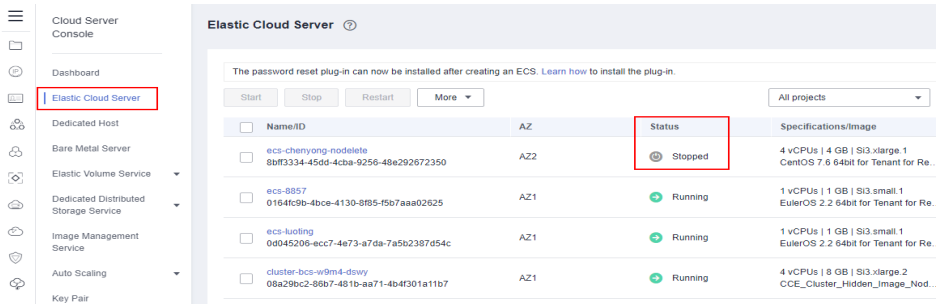
  If the cluster status is abnormal, locate the fault by following the CCE instructions: **How Do I Rectify the Fault When the Cluster Status Is Unavailable?**

  ii. Log in to the CCE console, click **Nodes**, and check the status of the abnormal blockchain's nodes.

If the node status is abnormal, locate the fault by following the CCE instructions: **What Should I Do If a Cluster Is Available But Some Nodes Are Unavailable?**

b. Check the ECS status.

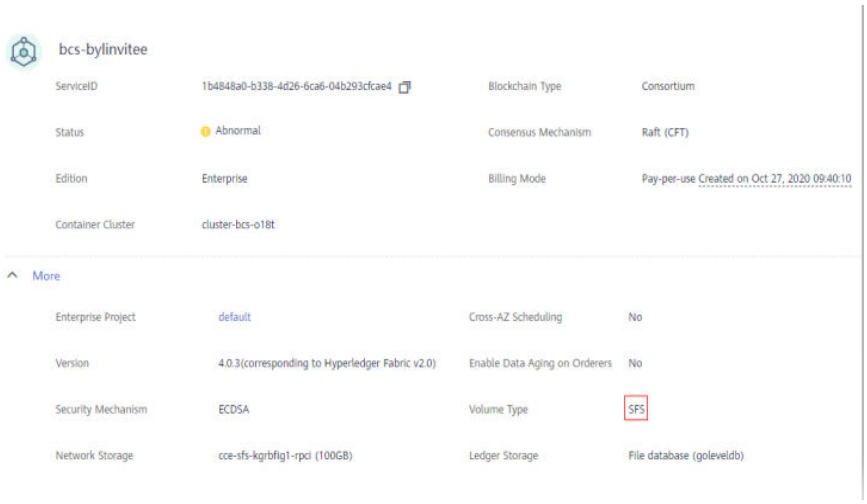
Log in to the ECS console and check the status of the ECS where the abnormal blockchain is deployed. ECS names usually take the following format: **Name of the target blockchain's cluster-A random number**.



If the ECS is in the **Stopped** state, start the ECS, wait for about 5 minutes, and check its status again.

c. Check the storage status.

i. Log in to the BCS console, go to the **Instance Management** page, and click the abnormal instance to view its volume type.



ii. Log in to the CCE console, click **Clusters**, and select a target cluster. On the cluster details page, click **Storage**. On the **PersistentVolumeClaims (PVCs)** tab page, check the storage status.

iii. If the SFS Turbo volume is in the **Lost** state, the state of the BCS instance will be displayed as **Abnormal**.

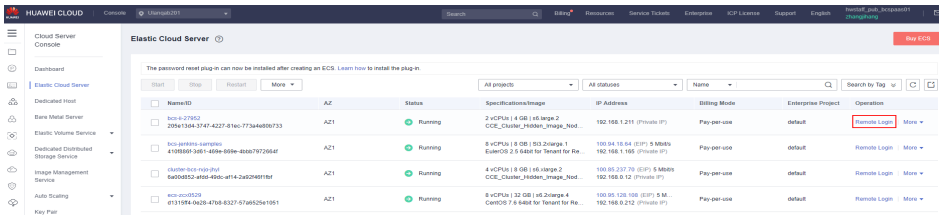
Solution:

Check whether the SFS Turbo file system exists or whether it is frozen, or contact the SFS technical support.

- Check item 2: Check whether the ECS specifications can meet the requirements.

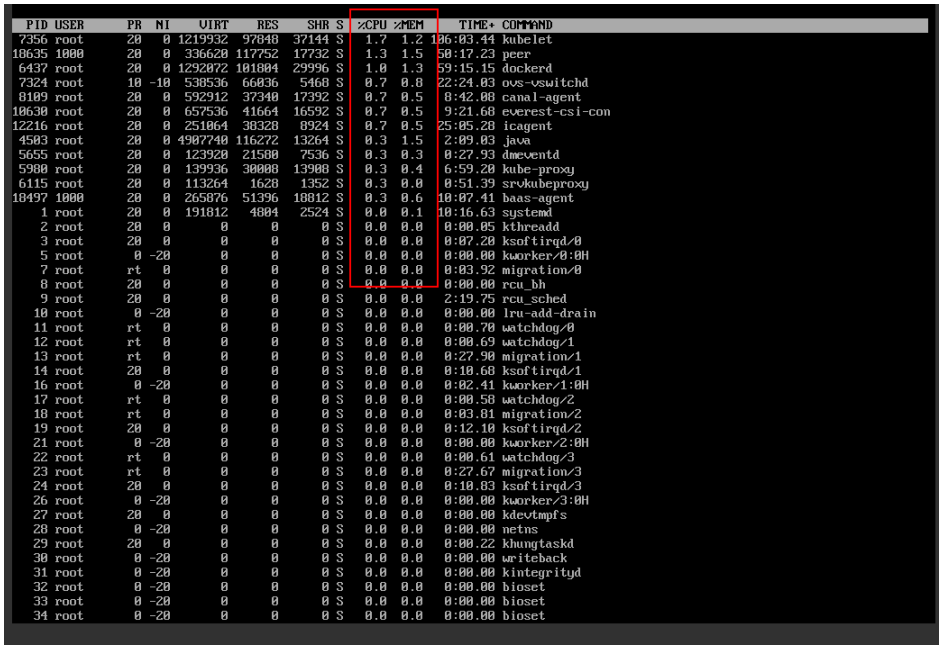
a. Log in to the ECS where the BCS instance is deployed.

Log in to the ECS console, locate the target ECS, and click **Remote Login** in the **Operation** column. ECS names usually take the following format: **Name of the target blockchain's cluster-A random number**.

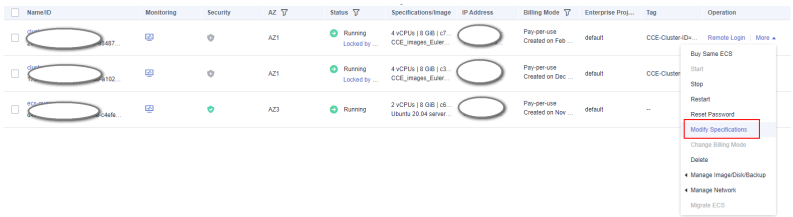


b. Run the **top** command to check whether the resource usage of any application is too high.

Figure 1-3 top command details



- If the CPU or memory usage of the peer, orderer, and baas-agent containers exceeds 60% and continues to increase as the transaction quantity increases, the current ECS specifications cannot meet the transaction requirements. In this case, you need to expand the ECS specifications.



- If there are resources taking up 100% or even higher CPU or memory usage, contact technical support to remove unnecessary resources.

1.2.4.2 What Can I Do If a BCS Instance Is in the Unknown State?

Symptom

The BCS instance is in the **Unknown** state.

Fault Locating

Check item 1: Check whether the cluster is hibernated.

Check item 2: Check whether the cluster exists.

Solution

- Check item 1: Check whether the cluster is hibernated.
  a. Log in to the BCS console, go to the **Instance Management** page, and click the abnormal instance to view its container cluster.
  b. Log in to the CCE console, click **Clusters**, and check the status of the target cluster.
  c. If the cluster is in the **Hibernating** state, the BCS instance will be in the **Unknown** state.
  d. Wake up the cluster. The BCS instance will become normal.
- Check item 2: Check whether the cluster exists.
  a. Log in to the BCS console, go to the **Instance Management** page, and click the abnormal instance to view its container cluster.
  b. Log in to the CCE console, click **Clusters**, and check the target cluster.
  c. If the cluster where the BCS instance is deployed does not exist, the BCS instance status is displayed as **Unknown**. If the cluster is not manually deleted, contact the CCE technical support.

1.2.4.3 What Can I Do If a BCS Instance Is in the EIP abnormal State?

Symptom

The BCS instance is in the **EIP abnormal** state.

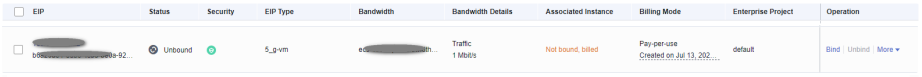
Fault Locating

Check item: Check whether the EIP has been unbound or released.

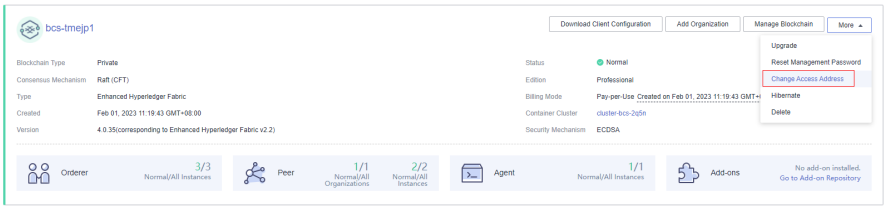
1. On the BCS console, click **Instance Management**. On an instance card, choose **More** > **Change Access Address** to check the EIP.
2. Go to the Network Console, locate the target EIP, and check its status.

Solution

1. If the EIP has been unbound, click **Bind** in the **Operation** column of the target EIP on the Network Console. Then, go back to the BCS console and refresh the **Instance Management** page.



2. If the EIP has been released, it is not displayed in the EIP list. In this case, purchase a new EIP and bind it. For details, see **Assigning an EIP**. Then, go back to the BCS console. On the **Instance Management** page, choose **More** > **Change Access Address** on an instance card. Select the target EIP and click **OK**.



3. If an ECS is imported to a cluster created earlier, and the ECS is not tagged, the EIP of the BCS instance will be abnormal.

   Solution: Log in to the ECS console, click a target ECS, on the **Tags** tab page, add a tag to the ECS instance. Set the tag key to **CCE-Dynamic-Provisioning-Node** and the tag value to any number. For details, see **Adding Tags**.

1.2.4.4 What Can I Do If a BCS Instance Is in the Frozen or Cluster frozen State?

Symptom

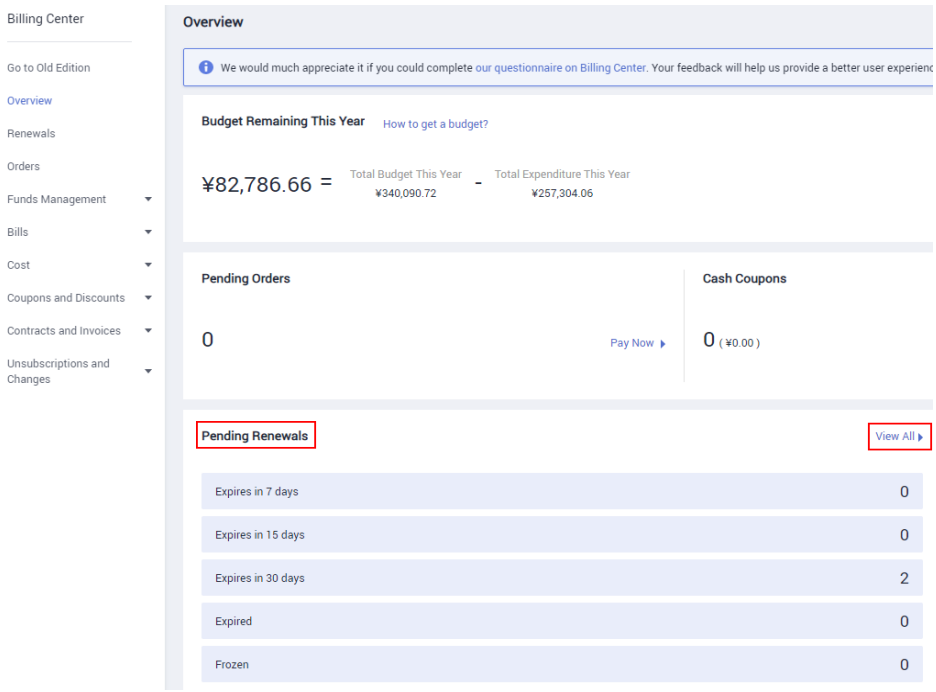
The BCS instance is in the **Frozen** or **Cluster frozen** state.

Fault Locating

Check whether the account is in arrears. If the account is in arrears, resources will be frozen.

Solution

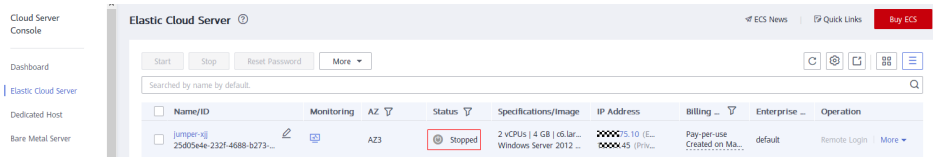
1. Log in to the BCS console and click **Billing Center** in the upper part of the page.
2. In the **Pending Renewal** area, click **View All**. On the displayed page, select the resources to be renewed and click **Renew**.



3. After the renewal is complete, go to the ECS console and check the status of the ECS where the BCS instance is deployed. (ECS names usually take the following format: **Name of the cluster where the BCS instance is deployed-A random number**.) If the ECS is in the **Stopped** state, wait for about 5 minutes and try again.

   NOTE

   After you renew the resources, the order status changes to **Completed**, indicating that the resources have been unfrozen and the resource status is restored.

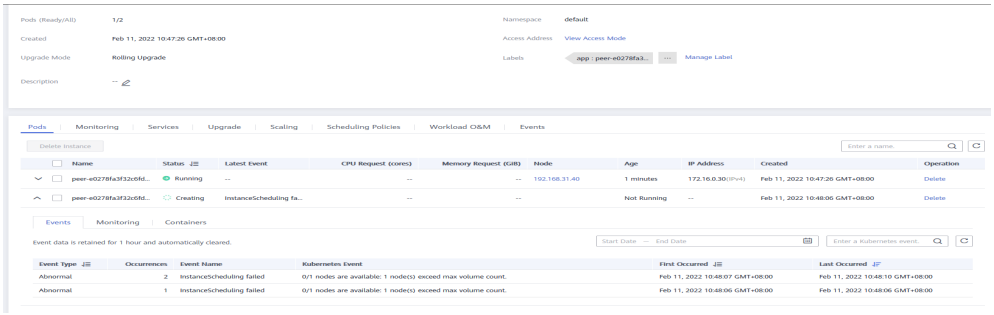


1.2.4.5 What Can I Do If the BCS Instance and the peer-xxx StatefulSet Are Abnormal After an Organization or a Peer Is Added?

Symptom

- More than 10 minutes after an organization or peer is added, the new pods remain abnormal. As a result, the change on the BCS instance times out and the instance status is abnormal. The task details show **BCS(XXX) wait for updating agent 400 times, stop updating** when an organization has been added or **wait the expand peer running exceed 100 times, stop waiting** when a peer has been added.
- Log in to the CCE console, choose **Workloads** > **StatefulSets**, select the cluster used by the BCS instance, and click the workload used by the new organization or peer. On the workload details page, view the pod list, and locate the abnormal pod. Check the pod events as the following figure shows.

Figure 1-4 Abnormal pods



Solution

Step 1 Log in to the CCE console, choose **Workloads** > **StatefulSets**, select the cluster used by the BCS instance, and click the workload used by the new organization or peer. On the workload details page, view the pod list, and locate the abnormal pod. Check the pod events to find out why the pod fails to be started.

NOTE

A maximum of 10 PVCs can be mounted to one CCE cluster node that is used by the current instance. When adding peers, use the following formula to obtain the number of required cluster nodes: Number of required cluster nodes = (number of new peers + number of the existing PVCs)/10. On the **Basic Information** page of an instance, click **View Details** next to **Network Storage**. Add numbers in the **SFS File System PVC** column to obtain the existing PVC quantity.

The possible cause is that the number of nodes in the current cluster has exceeded the upper limit.

Step 2 Click **Nodes**, select the cluster used by the BCS instance, click **Create Node** in the upper-right corner, and set parameters.

Step 3 Choose **Workloads** > **StatefulSets**, select the cluster used by the BCS instance, and click the workload used by the new organization or peer. On the workload details page, check the pod list and check whether the pods are normal.

- If yes, no further action is required.
- If no, contact technical support.

----End

1.2.5 Other Issues

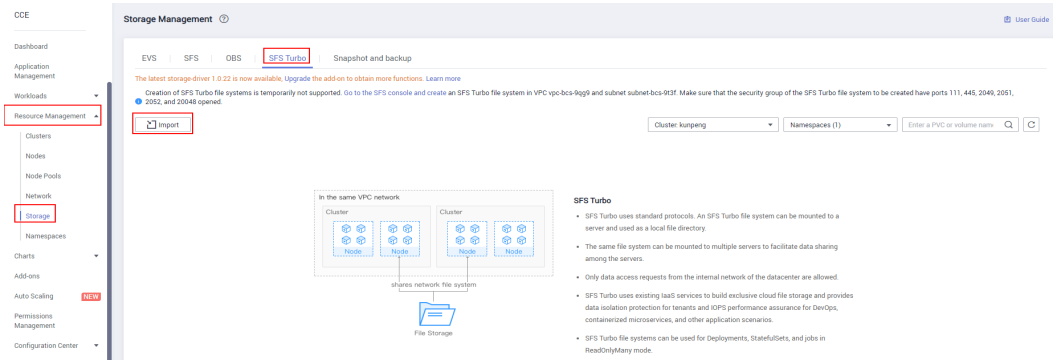
1.2.5.1 How Can I Enable Automatic Backup and Restore Data of an SFS Turbo File System?

Enabling Automatic Backup

If you select **SFS Turbo** for **Volume Type** and **Create SFS Turbo File System** for the network storage of a peer organization when purchasing a BCS instance, enable the automatic backup function of SFS Turbo on the SFS console. If this function is enabled, the system automatically backs up file system data on the

specified days. If file system data is deleted by mistake or contaminated, you can use the backup to restore the data, ensuring proper running of BCS.

After creating an SFS Turbo file system on the SFS console, import it on the CCE console before resuming BCS instance purchase. The following figure shows how to import an SFS Turbo file system.

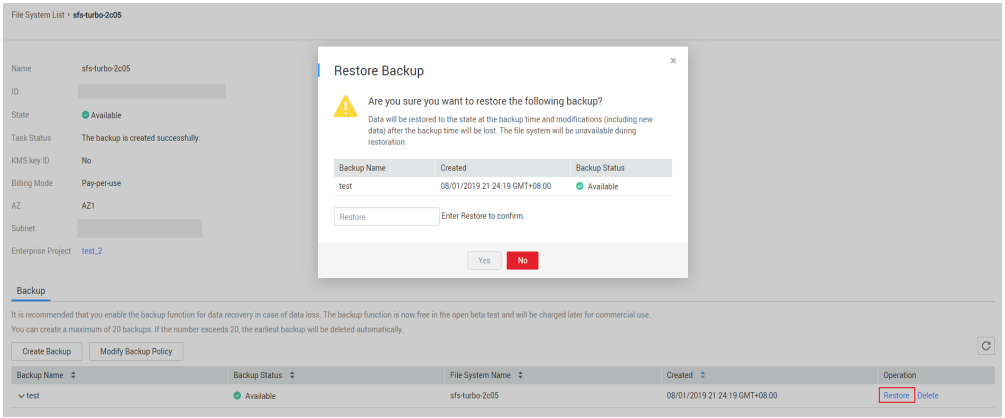


NOTE

If you select **SFS Turbo** for **Volume Type** and **Automatically create SFS Turbo file system** for the network storage of a peer organization, the automatic backup function is enabled by default, and the data is backed up at 02:00 every day.

Restoring Data

1. On the BCS console, choose **More** > **Hibernate** on a target instance card.
2. Go to the SFS console, click the SFS Turbo file system, locate the backup generated at the desired time, click **Restore** in the **Operation** column, and click **Yes**.



Wait until the data is restored. The file system is unavailable during data restore. After the restore is completed, the file system will become available again.

3. After the file system becomes available, go to the BCS console. On an instance card, choose **More** > **Wake** to wake up the instance.

NOTE

- Data will be restored to the state at the backup time, and the modifications and new data added after the backup time will be lost. The file system is unavailable during data restore.
- If a file system is deleted, its data cannot be restored from the backup.
- Each SFS Turbo file system supports a maximum of 20 backups. If the data is backed up when 20 backups exist, the earliest backup will be deleted. For details, see the backup function description of the SFS service.

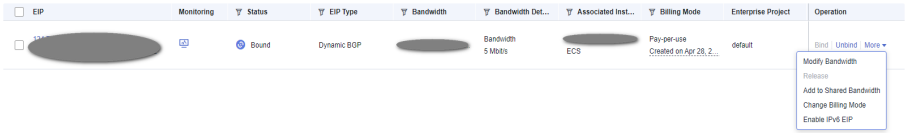
1.2.5.2 How Do I Enable the IPv6 EIP Function?

After the IPv6 EIP function is enabled, you will obtain an extra IPv6 EIP. External IPv6 addresses can access cloud resources through this IPv6 EIP. To enable the IPv6 EIP function, perform the following steps:

Step 1 Enable the IPv6 EIP function.

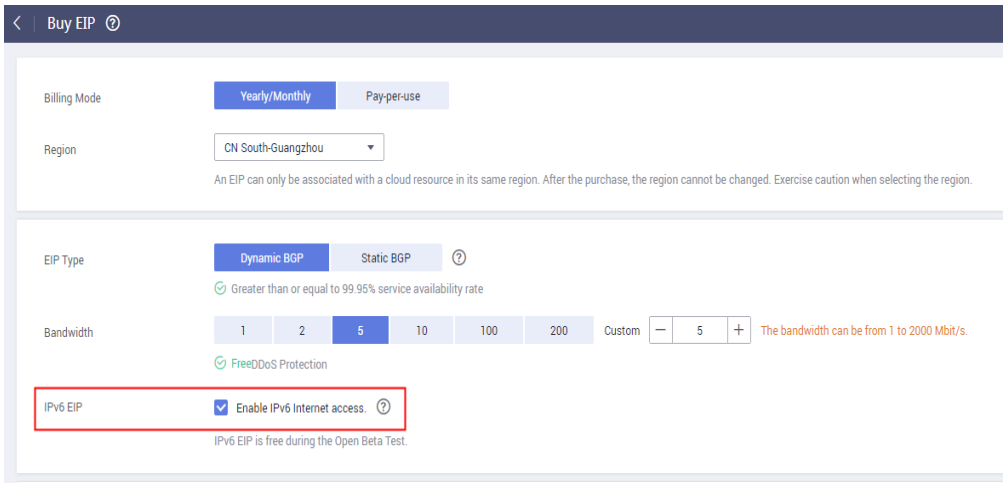
- For an existing EIP

  On the **EIPs** page of the Network Console, locate the target EIP, choose **More** > **Enable IPv6 EIP** in the **Operation** column.



- For a new EIP

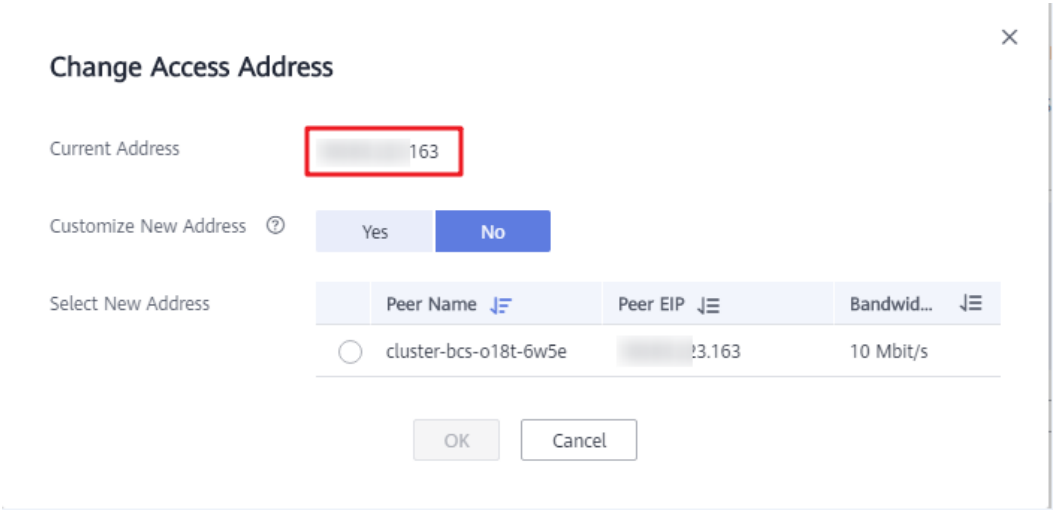
  On the **Buy EIP** page, select **Enable IPv6 Internet access.**



Step 2 Go to the BCS console, locate the BCS instance to which the EIP is bound, and choose **More** > **Change Access Address**. Confirm that the bound EIP is the same as the EIP for which the IPv6 EIP function has been enabled.

NOTE

This step is required for both existing and new BCS instances.



----End

1.2.5.3 What Can I Do If the Block Height Is Inconsistent Between Peers Due to Gossip Exceptions?

1. Run the following command to check the block height of the peer and then compare it with that of other peers. If there is a difference, block retrieving may have stopped or delayed.

   peer channel getinfo –c *{Channel Name}*
2. Restart the peer and obtain the blocks again. If the fault persists, proceed to steps **3** to **5**.
3. Go to the peer container. In the **/etc/hyperledger/fabric/** directory, configure the following parameters in the **core.yaml** file to synchronize blocks from the orderer to the peer:

   ```
   useLeaderElection: false
   orgLeader: true
   ```
4. Run the following command to query the process ID of **peer node start**:

   ps –ef
5. Run the following command to restart the peer process:

   kill -9 {pid}

1.3 Chaincode Management

1.3.1 How Do I Update a Chaincode If It Contains Bugs?

BCS supports chaincode upgrade. If you need to fix bugs in a chaincode, upload a new code package to update the chaincode.

1.3.2 How Do I View Chaincode Logs If My BCS Instance Uses Fabric v2.2?

Symptom

Chaincode container logs of a Fabric v2.2 BCS instance cannot be found on the AOM console.

Root Cause

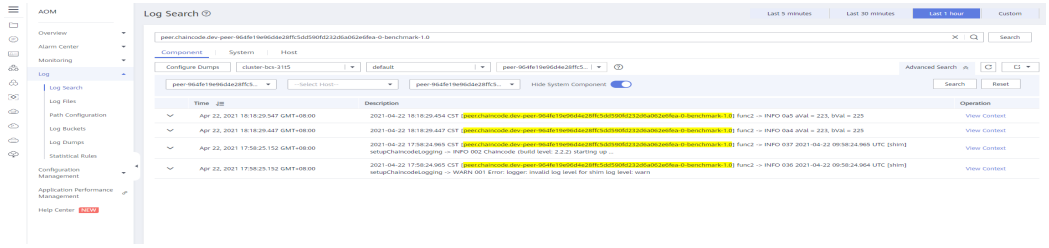
Currently, non-Fabric-v2.2 BCS instances use Kubernetes to start chaincode containers, whereas Fabric v2.2 BCS instances use Docker. The use of Docker is consistent with the open-source Hyperledger practice, and improves the stability of chaincode containers. On the AOM console, only the log files of the chaincode containers started using Kubernetes are displayed. Therefore, you cannot view the log files of chaincode containers started using Docker.

Solution

To enable users to view chaincode logs on the AOM console for troubleshooting, BCS outputs the run logs of a chaincode to the run logs of the peer where the chaincode is installed.

Search the peer logs for the keyword **[peer.chaincode.dev-peer-*Organization ID-Peer ID-Chaincode Name-Chaincode Version*]**. For example, you can search for the keyword **[peer.chaincode.dev-peer-964fe19e96d4e28ffc5dd590fd232d6a062e6fea-0-benchmark-1.0]** to find the corresponding chaincode log, as shown in the following figure:

Figure 1-5 Chaincode logs



1.3.3 What Can I Do If Decompression Failed During Chaincode Installation?

Symptom

The chaincode failed to be installed. A message is displayed, indicating that the chaincode package failed to be decompressed. The possible cause is that the format or content of the package is incorrect or the package does not contain a valid chaincode file.

Solution

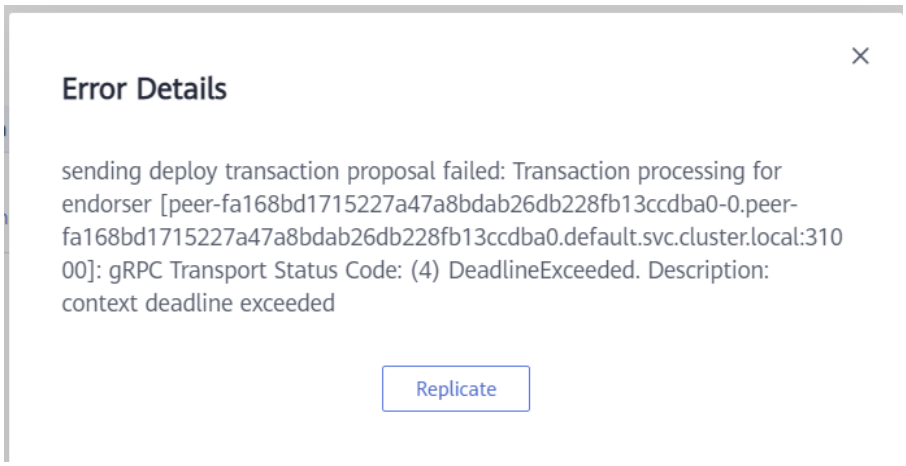
Check whether the chaincode package is in .zip format, if not, use WinRAR or packaging tools provided by Linux to compress it.

1.3.4 What Can I Do If "context deadline exceed" Is Displayed During Chaincode Instantiation?

Symptom

The chaincode fails to be instantiated, and the following message is displayed: **gRPC Transport Status Code: (4) DeadlineExceed. Description: context deadline exceed**.

Figure 1-6 Error details



Possible Causes

Chaincode compilation consumes resources. Each peer takes up 200 MB memory to compile a Go chaincode and 500–600 MB memory to compile a Java chaincode. If your VM flavor is insufficient, for example, less than 4 vCPUs and 8 GB memory, the compilation may time out.

Solution

1. Install and instantiate only one chaincode at a time.
2. Upgrade the VM flavor.
3. Instantiate the chaincode again. If the fault persists, contact technical support.

1.4 Data Storage to the Blockchain

1.4.1 What Can I Do When Transaction Connections Fail or Time Out?

Symptom

Transaction connections fail or time out.

Fault Locating

Check item 1: Check whether the transaction times out after the chaincode is instantiated for the first time.

Check item 2: Check whether the instance status is abnormal.

Check item 3: Check whether the Fabric SDK version used by the client matches the BCS instance version.

Check item 4: Check whether the ledger of the peer is updated.

Check item 5: Check whether the DB file exists.

Check item 6: If a BCS instance uses CouchDB of an earlier version for ledger storage, check whether BCS becomes unavailable after CouchDB restarts.

Check item 7: Check whether data can be stored to blockchain even though the request initiated from a blockchain client has timed out.

Solution

- Check item 1: Check whether the transaction times out after the chaincode is instantiated for the first time.

  During chaincode instantiation of a v3.0.x BCS instance (corresponding to Fabric v1.4.0), the container of only one peer in each organization is started. Other peer containers are built and started at the first transaction endorsement. This process takes a long time, and the transaction may time out.

  Figure 1-7 Transaction times out

  ```
  2021-01-29 11:45:43,703 WARN [org.hyperledger.fabric.sdk.helper.Config] - Failed to load any configuration from: config.properties. Using toolkit defaults
  2021-01-29 11:45:45,713 INFO [org.hyperledger.fabric.sdk.Channel] - Channel Channel{id: 1, name: channel} eventThread started shutdown: false  thread: null
  2021-01-29 11:46:20,913 ERROR [org.hyperledger.fabric.sdk.Channel] - Channel Channel{id: 1, name: channel} sending proposal with transaction a01db90518b5326e
  java.util.concurrent.TimeoutException: Waited 35000 milliseconds for io.grpc.stub.ClientCalls$GrpcFuture@50b0bc4c[status=PENDING, info=[GrpcFuture{clientCall
      at com.google.common.util.concurrent.AbstractFuture.get(AbstractFuture.java:471)
      at org.hyperledger.fabric.sdk.Channel.sendProposalToPeers(Channel.java:4108)
      at org.hyperledger.fabric.sdk.Channel.sendProposal(Channel.java:4032)
      at org.hyperledger.fabric.sdk.Channel.sendTransactionProposal(Channel.java:3581)
      at handler.FabricHelper.invokeBlockchain(FabricHelper.java:429)
      at handler.Main.LoopInvoke(Main.java:39)
      at handler.Main.main(Main.java:27)
  2021-01-29 11:46:20,916 ERROR [handler.FabricHelper] - Not enough endorsers for [a, b, 50]: 1. endorser error: Channel Channel{id: 1, name: channel} sending
  ```

  If you use a Go SDK, this problem does not need to be handled. The SDK will wait for the chaincode container to start, and no timeout will occur.

  If you use a Java SDK, you can set **req.setProposalWaitTime()**, the timeout for a request to endorse, in your application. In the following figure, the timeout is set to 60 seconds to avoid transaction failure after the chaincode is instantiated for the first time.

Figure 1-8 Transaction fails

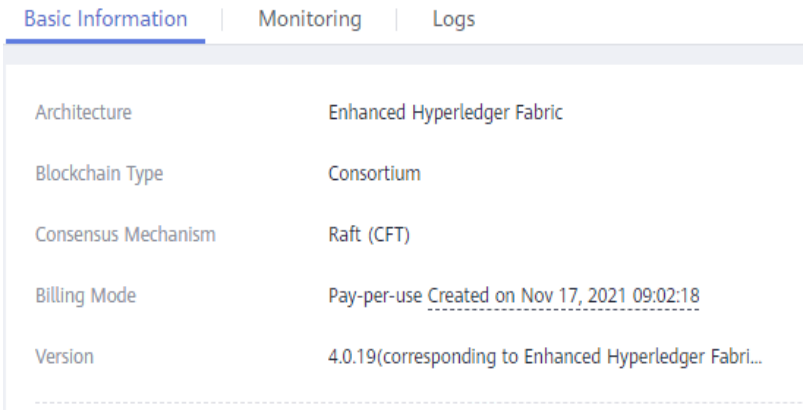
```
try {
    TransactionProposalRequest req = client.newTransactionProposalRequest();
    ChaincodeID cid = ChaincodeID.newBuilder().setName(chaincodeName).build();
    req.setChaincodeID(cid);
    req.setFcn(method);
    req.setArgs(args);

    Map<String, byte[]> tm2 = new HashMap<>();
    tm2.put("HyperLedgerFabric", "TransactionProposalRequest:JavaSDK".getBytes(UTF_8));
    tm2.put("method", "TransactionProposalRequest".getBytes(UTF_8));
    tm2.put("result", ":)".getBytes(UTF_8));  /// This should be returned see chaincode.
    req.setTransientMap(tm2);
    req.setProposalWaitTime(60000);
```

- Check item 2: Check whether the instance is abnormal.

  Log in to the BCS console and rectify the fault based on the instance status by following instructions provided in **What Can I Do If a BCS Instance Is in the Abnormal State?**

- Check item 3: Check whether the Fabric SDK version used by the client matches the BCS instance version.

  Log in to the BCS console, go to the **Instance Management** page, and click the abnormal instance to view its version.



  Record the value of **Version**. Check whether the Fabric SDK version used by the client is the same as the BCS version. If the versions are inconsistent, transactions may fail or time out.

  Solution

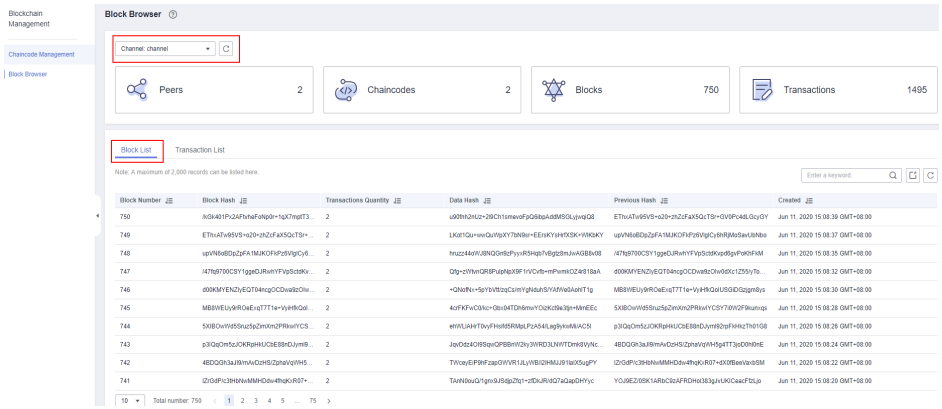
  Download the Fabric SDK of the version that corresponds to the Hyperledger Fabric version of BCS.

  Download link: **https://github.com/hyperledger/fabric**

- Check item 4: Check whether the ledger of the peer is updated.

  a. Log in to the BCS console. In the navigation pane, click **Instance Management**. On the card containing the abnormal instance, click **Manage Blockchain**. On the **Block Browser** page, select the abnormal channel, and view the block quantity on the **Block List** tab page.

Figure 1-9 Block List page



b. Log in to the ECS where the blockchain is deployed, run the **docker ps| grep k8s_peer** command to check the peer containers, and record the ID of the container where transactions time out.

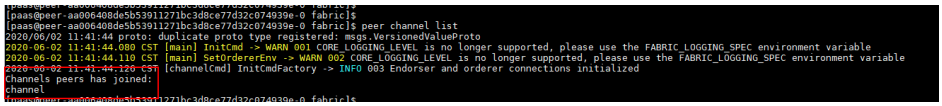
Figure 1-10 Checking the peer containers

c. Run the **docker exec -it** *Container ID* **/bin/bash** command to access the container.

Figure 1-11 Accessing the container

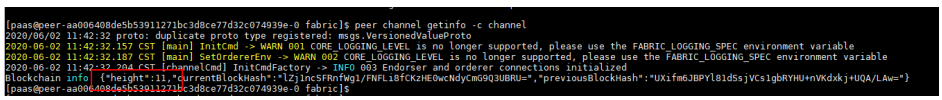
d. Run the **peer channel list** command to query the channel to which the peer is added.

Figure 1-12 Querying the channel to which the peer is added



e. Run the **peer channel getinfo -c** *{Channel name}* command to check whether the ledger of the peer is updated.

Figure 1-13 Checking whether the ledger of the peer is updated

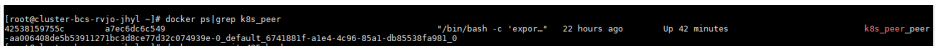


If the block quantity in the ledger of the peer is different from that displayed on the **Block Browser** page, query the block quantity again after 10 minutes. If the block quantity keeps unchanged, the ledger of the peer is not updated due to insufficient resources or high concurrency. As a result, transactions become abnormal.

f. If the problem persists, submit a service ticket. In the upper right corner of the console, choose **Service Tickets** > **Create Service Ticket**. Specify the issue as required.

- Check item 5: Check whether the DB file exists.
  a. Log in to the ECS where the blockchain is deployed, run the **docker ps| grep k8s_peer** command to check the peer containers, and record the ID of the container where transactions are abnormal.

     Figure 1-14 Checking the peer containers

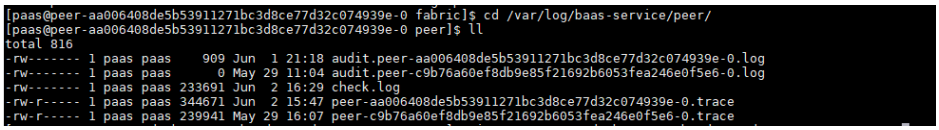
  b. Run the **docker exec -it** *Container ID* **/bin/bash** command to access the container.

     Figure 1-15 Accessing the container

  c. Run the **cd /var/log/baas-service/peer/** command to go to the directory where the logs of the peer are stored, and run the **ll** command to view all files.

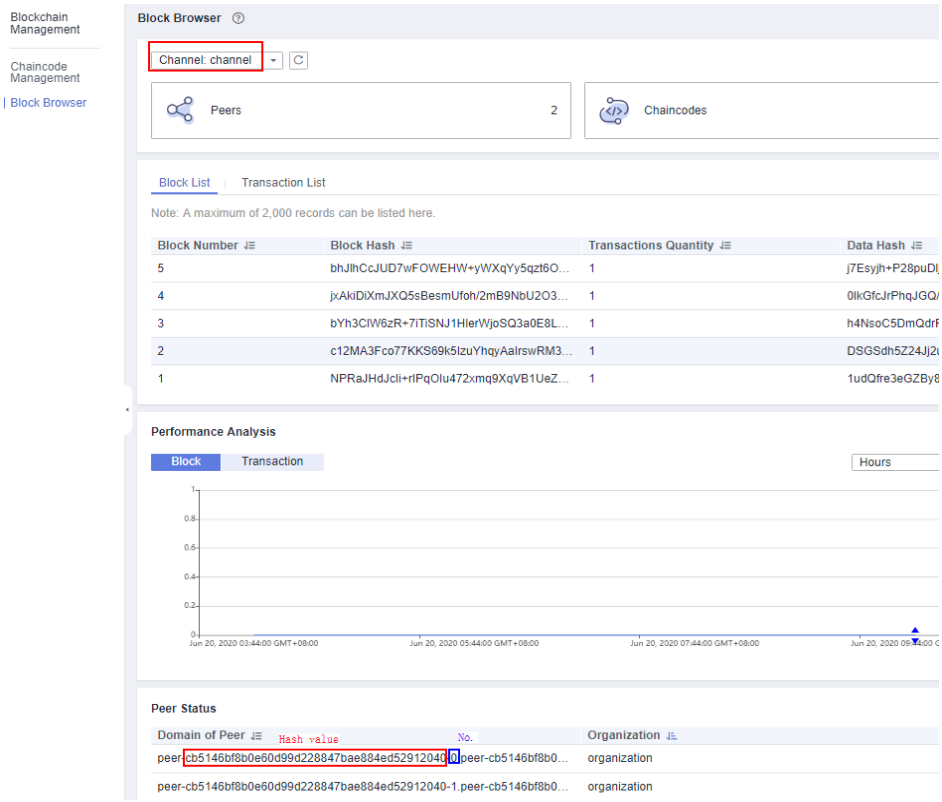
     Figure 1-16 Viewing all files

  d. Obtain the hash value and sequence number of the peer.

     On the **Block Browser** page of the Blockchain Management console, view the domain name and sequence number of the peer in the **Domain of Peer** column in the **Peer Status** list.

Figure 1-17 Peer Status list



e. The peer log file is named in the following format: **peer-***{Hash value}*-*{Serial number}*.**trace**. Run **cat** *{File name}*|**grep -C 5 "Fail to recover DB: file does not exist"** to search for exception information.

If you see **Fail to recover DB: file does not exist**, the DB file of the peer does not exist. As a result, transactions are abnormal.

f. If the problem persists, submit a service ticket. In the upper right corner of the console, choose **Service Tickets** > **Create Service Ticket**. Specify the issue as required.

- Check item 6: If a BCS instance uses CouchDB of an earlier version for ledger storage, check whether BCS becomes unavailable after CouchDB restarts.

  If an instance is of an earlier version and uses CouchDB for ledger storage, the status data will be lost after the instance is restarted, because the status data is not stored in the SFS file system. In this case, CouchDB reloads the block data to generate the status data, and the instance is unavailable for a certain period of time.

  It takes about 2 hours to synchronize data of 150,000 blocks. During data synchronization, port 7051 of the peer cannot be accessed.

  Solution:

  a. Upgrade the BCS instance to the latest version to avoid this problem when you perform upgrade or restart.

NOTE

When a BCS instance is upgraded to the latest version for the first time, the CouchDB container mounts the web disk and synchronizes status data. As a result, the BCS instance is unavailable for a certain period of time. This duration increases linearly as the block quantity increases. It takes about 2 hours to synchronize data for every 150,000 blocks. The block quantity can be viewed on the **Block Browser** page of the Blockchain Management console.

b. Log in to the BCS console. Choose **More** > **Upgrade** on the instance card.

c. In the dialog box that is displayed, view the current instance version or upgrade the BCS instance to the latest version.

NOTE

- Instances are unavailable during version upgrade. In a consortium, if your blockchain upgrades, all consortium blockchains must also upgrade. Reach an agreement with consortium members before you perform upgrade to eliminate effects on their blockchains.
- Do not initiate version upgrade when the chaincode is being installed or instantiated.
- You can upgrade BCS from the version corresponding to Hyperledger Fabric v1.4 to the version corresponding to Hyperledger Fabric v2.2. If a blockchain in the consortium has upgraded, all consortium blockchains must also upgrade to the same version to ensure successful transactions.
- BCS v4.x.x corresponds to Hyperledger Fabric v2.2.
- You can only upgrade BCS from an earlier version to a later version. Rollback is supported only if the upgrade fails.

- Check item 7: Check whether data can be stored to blockchain even though the request initiated from a blockchain client has timed out.

  If the error message **request timed out or been cancelled** is displayed on the client and **UTC is more than 15mos apart from current server time** is displayed in the organization node logs, ensure that the time and time zone of the client are the same as those of the organization node.

1.4.2 What Can I Do If the Network Connection Is Terminated or Rejected During Blockchain Access?

Increase the node bandwidth or host specifications, or reduce the transaction concurrency.

1.4.3 How Is Data Stored to the Blockchain?

Data is stored to the blockchain in the form of blocks. The block generation policy can be configured when you buy a BCS instance. For example, a new block can be generated every 1 second, when there are 500 transactions in the block, or when the block size reaches 2 MB, whichever condition is met first. For details about how to set block generation information, see **Deployment Using a CCE Cluster**.

1.4.4 How Is Data Synchronized Between Consortium Members?

Members of a consortium share a ledger. Except for privacy data, all transaction blocks and records are synchronized. Consortium members share orderers, and the

blocks on the peers of all participants are obtained from the orderers. Therefore, data is synchronized between consortium members in the unit of blocks. Cryptographic algorithms and consensus algorithms are used to keep block data consistent and immutable.

1.5 Demos and APIs

1.5.1 Demo Problems

1.5.1.1 General Checks

Fault Locating

Check item 1: Check whether the account is in arrears.

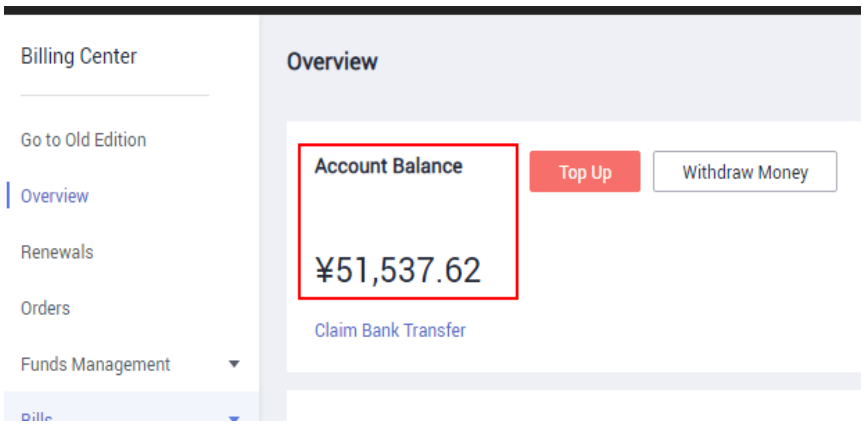
Check item 2: Check whether the instance status is normal.

Check item 3: Check whether the chaincode is properly instantiated.

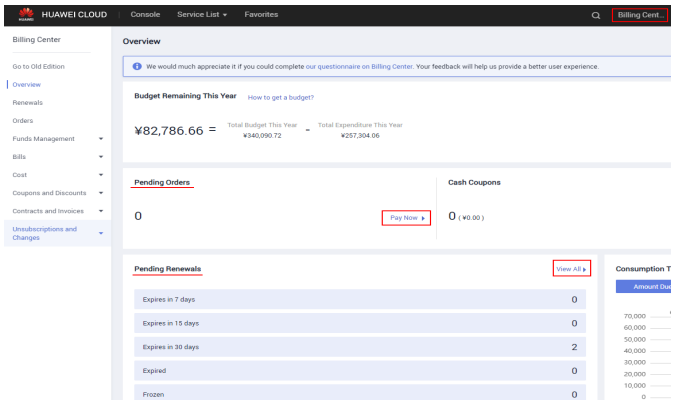
Check item 4: Check whether the demo has been modified.

Solution

- Check item 1: Check whether the account is in arrears.
  a. Log in to the BCS console and click **Billing Center** in the upper part of the page.



  If **Account Balance** is zero or less than the price displayed when you purchase the BCS instance, the purchase will fail.
  b. In the **Pending Orders** area, click **Pay Now**. On the **Orders** page that is displayed, select the order to be paid and complete the payment.



If the payment fails and a message is displayed indicating that the balance is insufficient, the purchase will also fail.

c. In the **Pending Renewal** area, click **View All**. On the displayed page, select the resources to be renewed and click **Renew**.

- Check item 2: Check whether the instance status is normal.

  Log in to the BCS console. In the navigation pane, click **Instance Management**. View the status of the target instance. For details, see **Abnormal Instance Statuses**.

- Check item 3: Check whether the chaincode is properly instantiated.

  a. Log in to the BCS console. In the navigation pane, click **Instance Management**. On the target instance card, click **Manage Blockchain**. Enter the password and verification code and click **Log In**. Then, go to the **Chaincode Management** page.

  b. Click **View more** in the **Instantiation Channel** column to view the instantiation status of the target chaincode.

- Check item 4: Check whether the demo has been modified.

  Follow the instructions in the next sections to check whether the specified demo has been modified.

1.5.1.2 Checking the Java SDK Application Demo

Fault Locating

1. Check whether the BCS instance is named **java-sdk-demo**, uses the Enterprise edition if the security mechanism is OSCCA-published cryptographic algorithms, is a private blockchain deployed in a CCE cluster, and uses SOLO for the consensus mechanism.

   NOTE

   Ensure that the development tools have been correctly installed and configured. For details, see **Preparation**.

   To check these details, log in to the BCS console, go to the **Instance Management** page, and click the target instance to view its details. For details about the configurations required for this demo, see **Java SDK Demo**.

2. Check whether the orderer and peer certificates have been downloaded, and whether the SDK file (**java-sdk-demo-sdk-config.zip**) has been downloaded to the **config** directory of the demo project. For details, see sections

"Download the SDK configuration" and "Download certificates" under **Configuring the Application**.

> **NOTE**
>
> The chaincode name specified when you set the SDK file parameters must be the same as that specified during chaincode installation.

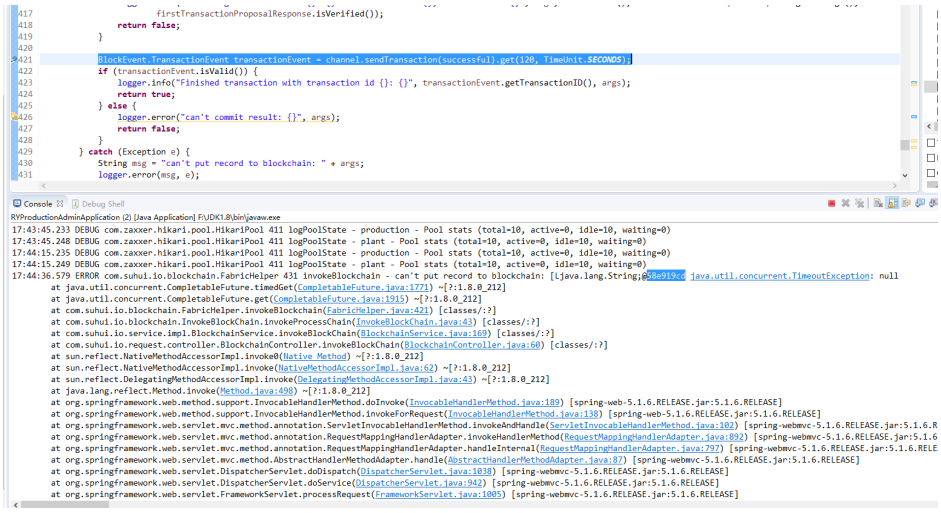
3. Check whether the path of the Main class has been changed as required. For details, see **Deploying the Application**.

   Find the **Main.java** file in the **/javasdkdemo/src/main/java/handler/** directory, and change the file path in the following code of the Main class to the absolute path of the **java-sdk-demo-sdk-config.yaml** file. Change the backslashes (\) to slashes (/).

   helper.setConfigCtx("E:/yourdir/huawei.yaml");

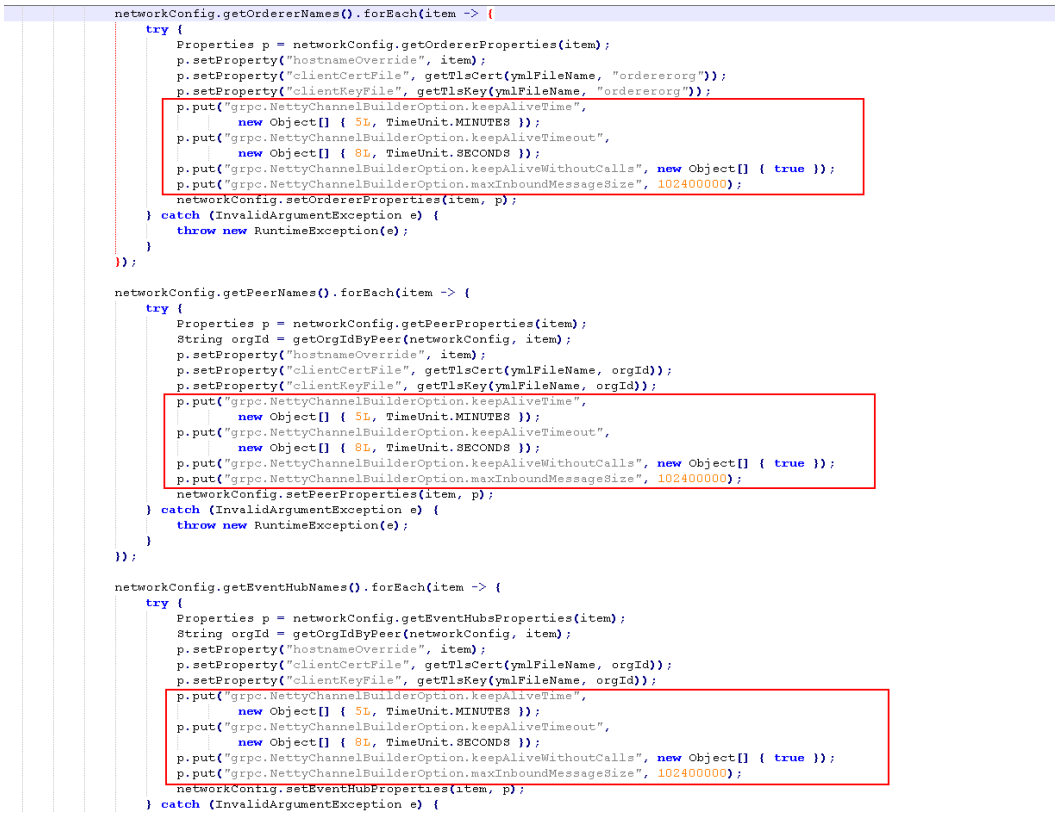
4. If the BCS instance uses OSCCA-published cryptographic algorithms, check whether the dependency upon **fabric-sdk-java-1.4.1-jar-with-dependencies.jar** is added to **pom.xml**. To add the dependency, remove the comments on the dependency as shown in the following figure.

5. If timeout occurs occasionally and the transaction interval is long, check whether the fault is caused by the connection protection mechanism of a third-party gRPC. The following figure shows the error information.



Solution

In the **src\main\java\handler\FabricHelper.java** file, add **p.put** to the sample code as shown in the following figure. Save the code and perform the transaction again.



```
networkConfig.getOrdererNames().forEach(item -> {
    try {
        Properties p = networkConfig.getOrdererProperties(item);
        p.setProperty("hostnameOverride", item);
        p.setProperty("clientCertFile", getTlsCert(ymlFileName, "ordererorg"));
        p.setProperty("clientKeyFile", getTlsKey(ymlFileName, "ordererorg"));
        p.put("grpc.NettyChannelBuilderOption.keepAliveTime",
                new Object[] { 5L, TimeUnit.MINUTES });
        p.put("grpc.NettyChannelBuilderOption.keepAliveTimeout",
                new Object[] { 8L, TimeUnit.SECONDS });
        p.put("grpc.NettyChannelBuilderOption.keepAliveWithoutCalls", new Object[] { true });
        p.put("grpc.NettyChannelBuilderOption.maxInboundMessageSize", 102400000);
        networkConfig.setOrdererProperties(item, p);
    } catch (InvalidArgumentException e) {
        throw new RuntimeException(e);
    }
});

networkConfig.getPeerNames().forEach(item -> {
    try {
        Properties p = networkConfig.getPeerProperties(item);
        String orgId = getOrgIdByPeer(networkConfig, item);
        p.setProperty("hostnameOverride", item);
        p.setProperty("clientCertFile", getTlsCert(ymlFileName, orgId));
        p.setProperty("clientKeyFile", getTlsKey(ymlFileName, orgId));
        p.put("grpc.NettyChannelBuilderOption.keepAliveTime",
                new Object[] { 5L, TimeUnit.MINUTES });
        p.put("grpc.NettyChannelBuilderOption.keepAliveTimeout",
                new Object[] { 8L, TimeUnit.SECONDS });
        p.put("grpc.NettyChannelBuilderOption.keepAliveWithoutCalls", new Object[] { true });
        p.put("grpc.NettyChannelBuilderOption.maxInboundMessageSize", 102400000);
        networkConfig.setPeerProperties(item, p);
    } catch (InvalidArgumentException e) {
        throw new RuntimeException(e);
    }
});

networkConfig.getEventHubNames().forEach(item -> {
    try {
        Properties p = networkConfig.getEventHubsProperties(item);
        String orgId = getOrgIdByPeer(networkConfig, item);
        p.setProperty("hostnameOverride", item);
        p.setProperty("clientCertFile", getTlsCert(ymlFileName, orgId));
        p.setProperty("clientKeyFile", getTlsKey(ymlFileName, orgId));
        p.put("grpc.NettyChannelBuilderOption.keepAliveTime",
                new Object[] { 5L, TimeUnit.MINUTES });
        p.put("grpc.NettyChannelBuilderOption.keepAliveTimeout",
                new Object[] { 8L, TimeUnit.SECONDS });
        p.put("grpc.NettyChannelBuilderOption.keepAliveWithoutCalls", new Object[] { true });
        p.put("grpc.NettyChannelBuilderOption.maxInboundMessageSize", 102400000);
        networkConfig.setEventHubProperties(item, p);
    } catch (InvalidArgumentException e) {
```

p.put("grpc.NettyChannelBuilderOption.keepAliveTime", new Object[] {5L, TimeUnit.MINUTES});

p.put("grpc.NettyChannelBuilderOption.keepAliveTimeout", new Object[] {8L, TimeUnit.SECONDS});

p.put("grpc.NettyChannelBuilderOption.keepAliveWithoutCalls", new Object[] {true});

p.put("grpc.NettyChannelBuilderOption.maxInboundMessageSize", 102400000);

1.5.1.3 Checking the RESTful API Application Demo

Fault Locating

1. Check whether the BCS instance is named **demo**, uses the Professional edition, is a private blockchain deployed in a CCE cluster, uses ECDSA for the security mechanism, and has enabled support for RESTful APIs. If support for RESTful APIs was not enabled during BCS instance creation, you can still enable it by installing an add-on after the BCS instance is created.

   To check these details, log in to the BCS console, go to the **Instance Management** page, and click the target instance to view its details. For details about the configurations required for this demo, see **REST API Demo**.

2. Check whether the user certificate has been downloaded and whether parameters in **conf.yaml** and **main.go** have been modified.

   Download the user certificate by referring to **Downloading SDK Configurations and Certificates**.

For details on how to modify the parameters in the **conf.yaml** and **main.go** files, see section "Modify parameter settings" under **Configuring the Application**.

1.6 O&M and Monitoring

1.6.1 How Do I Clear Residual Log Files After a BCS Service Is Deleted?

After a BCS service is deleted, log files are not automatically deleted from the cluster nodes. You are advised to manually delete the residual files to save space.

Use the remote management tool to log in to each cluster node used by the deleted BCS service, and check whether there are residual log files in the following paths:

```
/var/paas/sys/log/baas-agent
/var/paas/sys/log/baas-restapi
/var/paas/sys/log/baas-service
```

If residual log files exist, run the following command to delete them:

```
rm -rf /var/paas/sys/log/baas-agent /var/paas/sys/log/baas-restapi /var/paas/sys/log/baas-service
```

1.6.2 Why Is "TLS handshake failed" Repeatedly Displayed in the Instance Log?

Symptom

The error message "TLS handshake failed" is repeatedly displayed in the instance log.

Root Cause

The inviting party has dismissed the consortium and created a new BCS instance. However, the BCS instance of the invitee still exists and repeatedly sends incorrect network requests to the new BCS instance of the inviting party.

Solution

The inviting party changes the EIP or creates another BCS instance in a different CCE cluster.

1.7 Consortium Management

1.7.1 Can I Invite Individual Users to Join a Consortium?

Yes. You can invite any users (individual and enterprise users) of the cloud service platform to join a consortium.

1.8 Agency Permissions

1.8.1 How Do I Adjust Agency Permissions of My Service?

Due to account security concerns, unnecessary agency permissions should be deleted for your enhanced Hyperledger Fabric instances.

The procedure is as follows.

Step 1 Log in to the IAM console, click **Agencies**, and search for **bcs_admin_trust**.

Step 2 Click **Modify**.

Step 3 Go to the **Permissions** tab page and confirm that the following permissions have been correctly configured:

- Configuration of **BCS agency policy for global**:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "obs:object:DeleteObject",
            "obs:bucket:HeadBucket",
            "obs:object:PutObject",
            "obs:bucket:CreateBucket"
         ]
      }
   ]
}
```

- Configuration of **BCS agency policy for project**:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Action": [
            "cce:cluster:get",
            "cce:node:list",
            "ecs:cloudServers:listServerInterfaces",
            "ecs:cloudServers:list",
            "ecs:serverInterfaces:get",
            "vpc:publicIps:list",
            "vpc:publicIps:update",
            "vpc:ports:get",
            "cbr:vaults:delete",
            "cbr:vaults:get",
            "cbr:vaults:removeResources",
            "cbr:vaults:addResources",
            "cbr:vaults:create",
            "cbr:vaults:update",
            "cbr:vaults:backup",
            "cbr:backups:delete",
            "cbr:backups:list",
            "evs:volumes:list",
            "sfsturbo:shares:getShare"
         ],
         "Effect": "Allow"
      }
   ]
}
```


Step 4 Delete the following unnecessary permissions:

- SFS Administrator
- ECS FullAccess
- VPC Administrator
- EPS FullAccess
- CCE Administrator
- AOM FullAccess
- APM FullAccess
- OBS Administrator
- CBR FullAccess
- BCS Administrator

----End

