

## Blockchain Service

# User Guide

**Issue** 01  
**Date** 2023-10-11



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2023. 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

---

<b>1 Service Overview.....</b>	<b>1</b>
1.1 What Is BCS?.....	1
1.2 Functions.....	2
1.3 Advantages.....	6
1.4 Key Concepts.....	7
1.5 Edition Differences.....	9
1.6 Application Scenarios.....	11
1.6.1 Transactions Between Subsidiaries.....	11
1.6.2 Supply Chain Logistics.....	13
1.6.3 Healthcare.....	15
1.7 Permissions Management.....	16
1.8 Billing.....	18
1.9 Restrictions.....	18
<b>2 Managing Enhanced Hyperledger Fabric Instances.....</b>	<b>19</b>
2.1 BCS Overview.....	19
2.2 Permissions Management.....	21
2.2.1 Creating a User and Granting BCS Permissions.....	21
2.2.2 Creating a Custom Policy.....	22
2.3 Instance Deployment.....	24
2.3.1 Deployment Using a CCE Cluster.....	24
2.4 Instance Management.....	31
2.4.1 Basic Operations.....	31
2.4.2 O&M Center.....	35
2.4.2.1 Viewing Monitoring Data and Logs.....	35
2.4.2.2 Viewing Alarms.....	37
2.4.2.3 Setting Web Disk Space Alarms.....	50
2.4.2.4 Disk Metrics.....	51
2.4.2.5 Viewing O&M Logs.....	52
2.4.2.6 Viewing Chaincode Debug Logs.....	55
2.5 Channel Management.....	56
2.6 Blockchain Management.....	58
2.6.1 Chaincode Management.....	58
2.6.2 Block Browser.....	66

2.7 Downloading SDK Configurations and Certificates.....	67
2.8 Consortium Management.....	69
2.8.1 Forming a Consortium.....	69
2.8.2 Member Management.....	70
2.8.3 Notification Management.....	71
2.9 Add-on Management.....	71
2.9.1 Add-on Overview.....	71
2.10 Contract Repository.....	73
<b>3 FAQs.....</b>	<b>75</b>
3.1 BCS FAQs.....	75
3.1.1 Instance Management.....	75
3.1.1.1 Consultation.....	75
3.1.1.1.1 How Do I Determine Whether a Blockchain Is Necessary?.....	75
3.1.1.1.2 What Underlying Framework Is Used for BCS?.....	75
3.1.1.1.3 What Competitive Advantages Does BCS Have?.....	76
3.1.1.1.4 What Are the Specifications of VMs to Be Created for BCS?.....	76
3.1.1.1.5 What Are the Differences Between Channel Isolation and Privacy Protection?.....	76
3.1.1.1.6 How Well Does BCS Perform?.....	76
3.1.1.1.7 When Do I Need to Hibernate or Wake an Instance?.....	77
3.1.1.2 Service Usage.....	77
3.1.1.2.1 How Do I Check Whether the ICAgent Is Installed for the Cluster?.....	77
3.1.1.2.2 What Can I Do If I Can't Open the Blockchain Management Console?.....	77
3.1.1.2.3 What Should I Do If My BCS Instance Remains in the Creating State?.....	78
3.1.1.2.4 What Should I Do If a Peer Restarts Frequently with the Error Message "PanicDB not exist"?.....	78
3.1.1.2.5 What Can I Do If the CPU Usage of a Blockchain Node Reaches 100%?.....	78
3.1.1.2.6 Why Can't I Log In to the Blockchain Management Console?.....	78
3.1.1.2.7 BCS.4009100: System Error.....	80
3.1.1.2.8 How Can I Obtain Private Keys and Certificates for Enhanced Hyperledger Fabric Blockchains?.....	81
3.1.1.2.9 Can All Blocks Be Saved As More and More Blocks Are Created?.....	82
3.1.1.3 Abnormal Instance Statuses.....	82
3.1.1.3.1 What Can I Do If a BCS Instance Is in the Abnormal State?.....	83
3.1.1.3.2 What Can I Do If a BCS Instance Is in the Unknown State?.....	84
3.1.1.3.3 What Can I Do If a BCS Instance Is in the EIP abnormal State?.....	85
3.1.1.3.4 What Can I Do If the BCS Instance and the peer-xxx StatefulSet Are Abnormal After an Organization or a Peer Is Added?.....	86
3.1.1.4 Other Issues.....	87
3.1.1.4.1 How Can I Enable Automatic Backup and Restore Data of an SFS Turbo File System?.....	87
3.1.1.4.2 What Can I Do If the Block Height Is Inconsistent Between Peers Due to Gossip Exceptions?.....	88
3.1.2 Chaincode Management.....	88
3.1.2.1 How Do I Update a Chaincode If It Contains Bugs?.....	88
3.1.2.2 How Do I View Chaincode Logs If My BCS Instance Uses Fabric v2.2?.....	88
3.1.2.3 What Can I Do If Decompression Failed During Chaincode Installation?.....	89

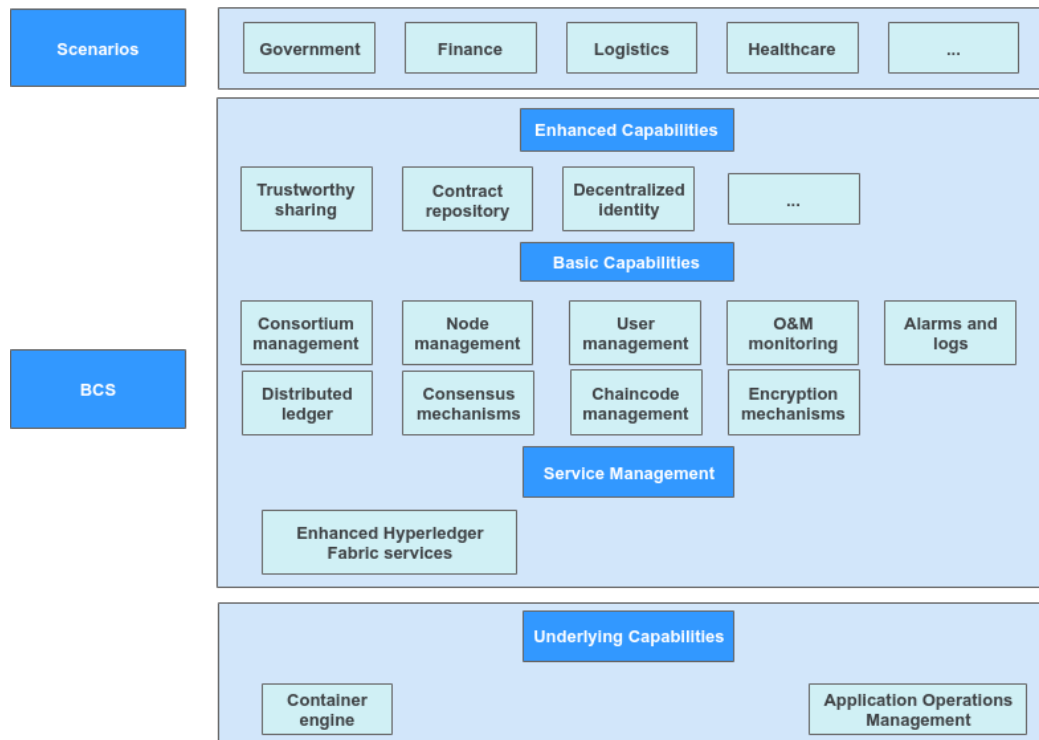
3.1.2.4 What Can I Do If "context deadline exceed" Is Displayed During Chaincode Instantiation?.....	89
3.1.3 Data Storage to the Blockchain.....	90
3.1.3.1 What Can I Do When Transaction Connections Fail or Time Out?.....	90
3.1.3.2 What Can I Do If the Network Connection Is Terminated or Rejected During Blockchain Access? .....	94
3.1.3.3 How Is Data Stored to the Blockchain?.....	94
3.1.3.4 How Is Data Synchronized Between Consortium Members?.....	94
3.1.4 Demos and APIs.....	94
3.1.4.1 Demo Problems.....	94
3.1.4.1.1 General Checks.....	95
3.1.5 O&M and Monitoring.....	95
3.1.5.1 How Do I Clear Residual Log Files After a BCS Service Is Deleted?.....	95
3.1.5.2 Why Is "TLS handshake failed" Repeatedly Displayed in the Instance Log?.....	96
3.1.6 Consortium Management.....	96
3.1.6.1 Can I Invite Individual Users to Join a Consortium?.....	96
<b>4 Change History.....</b>	<b>97</b>

# 1 Service Overview

## 1.1 What Is BCS?

Blockchain Service (BCS) is a blockchain platform for enterprises and developers. BCS helps you quickly deploy, manage, and maintain blockchain networks, lowering the threshold for using blockchains. In this way, you can focus on the development and innovation of your own business to quickly implement business using blockchains.

Figure 1-1 BCS architecture



- Infrastructure

The infrastructure layer offers underlying resources required for creating a blockchain network, including resources on nodes used to compute and store data in the network.

- **BCS**  
BCS provides enhanced Hyperledger Fabric blockchain instances, which consist of user management, node management, and O&M monitoring modules. It helps you quickly create, manage, and efficiently maintain an enterprise-grade blockchain system for upper-layer applications.
  - Enhanced Hyperledger Fabric instances are seamlessly integrated with Hyperledger Fabric, and are enhanced with the full-stack, trustworthy capabilities, including elastic computing, container, security, and AI services. They can meet enterprise- and finance-grade reliability, performance, and privacy requirements.
- **Scenarios**  
BCS can be used in multiple scenarios of various industries, such as supply chain finance, supply chain sourcing, digital assets, and notarization for crowdsourcing. Industry-specific applications connect to the blockchain platform to ensure data reliability and security.
- **Security management**  
Privacy isolation, consensus algorithms, and OSCCA-published cryptographic algorithms based on light nodes provide secure computing, trustworthy data sharing, and distributed identity capabilities.

## Benefits of Blockchain

**Higher efficiency:** Builds a trusted multi-party collaboration platform to reduce disputes and improve transaction efficiency.

**Reduced costs:** Reduces extra costs and the participation of third parties.

**Lower risks:** Precludes the possibility of tampering to reduce risks of frauds and network errors.

**Stronger trust:** Builds up trust between transaction participants using shared ledgers, processes, and records.

**Transparent audit:** Audit institutions can audit the immutable ledgers at any time.

## More Information

- Data in a blockchain system is generated and stored in blocks, which are chained in a time sequence. Hence the term "blockchain".
- All nodes in a blockchain system participate in data verification, storage, and maintenance. Consensus must be reached to create a block. Any new block is broadcast to all nodes, ensuring synchronization on the entire network. After this, it cannot be modified or deleted.

## 1.2 Functions

BCS provides the following functions to help you quickly deploy blockchains featuring security, high efficiency, and cost-effectiveness.

## Instance Deployment

You can create resources when deploying a blockchain system, without a need to prepare resources required by the system in advance.

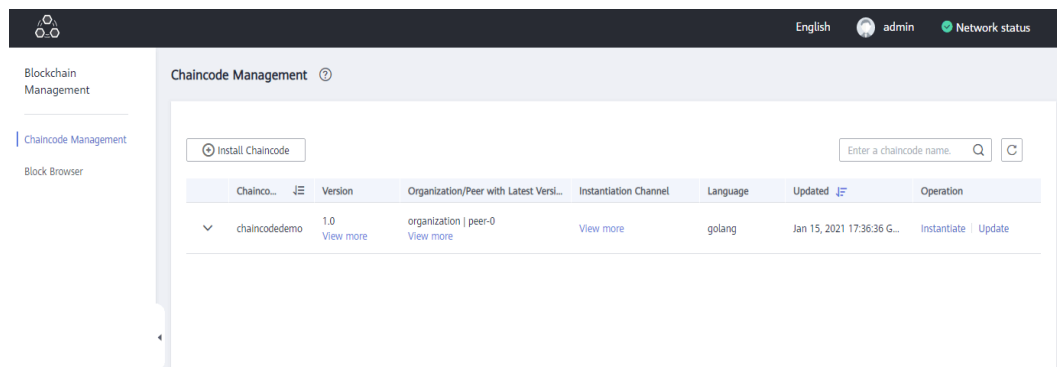
- The blockchain network configuration and deployment are completed in minutes, instead of days.
- Underlying technological details are masked. You do not need to care about the underlying technology implementation and platform construction.
- You can create consortium or private blockchains.

## Instance Management

You can view the running statuses of your BCS instances and perform operations on them, for example, adding organizations, upgrading, and obtaining client configurations.

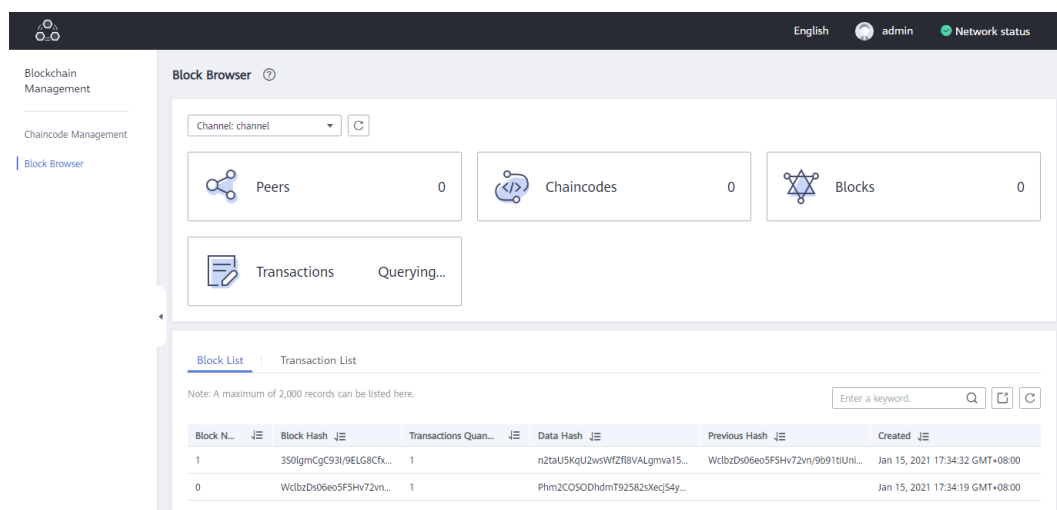
## Chaincode Management

You can manage chaincodes on the graphical user interface (GUI) throughout the entire chaincode lifecycle, including installation, instantiation, and upgrade.



## Block Browser

In the block browser, you can query the block and transaction quantities and details, peer statuses, and performance data for blockchain maintenance.





## Ledger Storage

File database (GoLevelDB) and NoSQL (CouchDB) are available for ledger storage.

- File database: Historical transaction data is stored in the blockchain, and status data is stored in LevelDB.
- NoSQL: Transaction and status data are stored in CouchDB.

Volume Type **Recommended**  
SFS Turbo  
SFS Turbo provides low-latency, high-IOPS file storage.

Storage Capacity of Peer Organization (GB) - 500 +

Ledger Storage File database (GoLevelDB) NoSQL (CouchDB)

Peer Organization Create peer organizations and specify the number of peers for each organization.

Peer Organization	Peers	Operation
organization	- 2 +	Delete

⊕ Add Peer Organization

Channel Configuration

Channel	Organization	Description	Operation
channel	organization <a href="#">↗</a>	<a href="#">↗</a>	Delete

⊕ Add Channel

## Consensus Algorithms

BCS supports two consensus algorithms for different scenarios.

- **Raft (CFT)**: A crash fault tolerance (CFT) algorithm that tolerates faults at a maximum of  $(N - 1)/2$  orderers, where N indicates the total number of orderers. It also supports Fabric v2.2.
- **FBFT**: The fast Byzantine fault tolerance (FBFT) algorithm. It requires 4 to 10 orderers for transaction ordering and tolerates faults at a maximum of  $(N - 1)/3$  orderers, where N indicates the total number of orderers. It also supports Fabric v2.2.

Consensus Mechanism

### Raft (CFT)

Tolerates faults at a maximum of  $(N - 1)/2$  orderers, where N indicates the total number of orderers.  
Supports Enhanced Hyperledger Fabric v2.2.

### FBFT

Requires 4 to 10 orderers.  
Tolerates faults at a maximum of  $(N - 1)/3$  orderers, where N indicates the total number of orderers.  
Supports Enhanced Hyperledger Fabric v2.2.

## Consortium Member and Organization Management

- A consortium initiator can dynamically invite other tenants to conveniently and quickly set up a consortium blockchain.
- You can dynamically add peer organizations to a BCS instance.

### Invite Tenant ?

i Ensure that the account name is correct. You can check the account name on the [My Credentials](#) page.

Service

Consortium Channel

\* Invitee

+ Add Tenant

## Auto Scaling of Nodes

You can scale nodes as required, without rebooting systems.

< Add Organization

**Note**

- Do not perform operations on the instance when adding organizations.
- After adding the organization to an existing channel, update the endorsement policy of the channel before instantiating the chaincode. Otherwise, certificates may fail the verification, causing an instantiation failure.

**Current Configuration**

Instance Name	bcs-9t5qw3	Instance ID	[Redacted]
Current Specification	1 instances   2 peers	Region	[Redacted]

**New Organization** Create peer organizations and specify the number of peers for each organization.  
A newly created SFS Turbo file system becomes available only after being imported on the Storage Management page of the CCE console.

i If the redirection fails, go to the old CCE console. Choose Resource Management > Storage, then click Create SFS File System to create a PVC. If you have any questions, contact CCE or BCS. X

Peer Organization	Peers	Network Storage	Operation
<input type="text" value="org1"/>	- 2 +	<input type="text" value="cce-efs-import-fr33bymw-mesa (500GB)"/>	C Delete

Add Peer Organization The number of peer organizations has reached the maximum allowed limit (1)

**New Specifications** 1 instances | 4 peers

## Contract Scan

Automatic analysis tools are provided to ensure the smart contract safety from the source. Based on the vulnerabilities and issues commonly found in consortium blockchain smart contracts, the check reports and solutions are generated to help users and developers audit code security, detect risks, and resolve problems.

## Privacy Protection

- In each channel, members are assigned different access permissions to certain data, ensuring the data privacy of members within a channel.
- Different channels are also isolated from each other, protecting block data of all members in a channel from other channels.

## Application Access

Applications can access blockchain networks using software development kits (SDKs) and RESTful APIs.

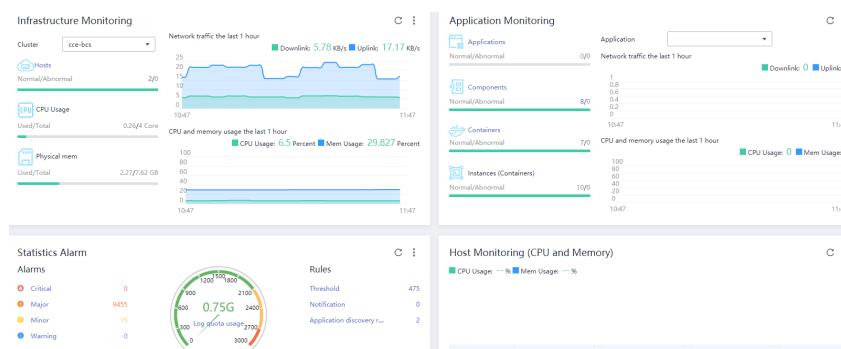
- SDK configuration files can be downloaded. After simple configuration, an application can be connected to a blockchain network.

- Applications can invoke chaincodes through RESTful APIs. The policy of multi-organization endorsement is supported.

## Monitoring and O&M

BCS connects to the monitoring platform to monitor data and resources in real time and generate alarms and notifications when necessary.

- Automated O&M: BCS actively upgrades the underlying blockchain platform and updates patches to seamlessly integrate with the O&M system.
- Enterprise-grade monitoring: Multi-dimensional monitoring is performed on clusters 24/7, and user-defined alarms can be reported through multiple channels.



## 1.3 Advantages

### Open and Easy to Use

Building an enterprise-grade distributed blockchain network is not easy. It requires not only in-depth knowledge of blockchain but also complex design and configuration, which is error-prone and costly.

- BCS can help enterprises deploy blockchain networks within only 5 minutes, reducing the development and deployment costs by as much as 80%.
- BCS hosts functions of full-lifecycle management and GUI-based smart contract coding, commissioning, and deployment. Customers using BCS can focus on the innovation and development of their own service applications.

### Flexible and Efficient

- BCS supports multiple efficient consensus algorithms and deeply optimizes existing algorithms to achieve balance between security and efficiency.
- Consensus within seconds can realize 100,000 transactions per second (TPS), meeting service performance requirements.
- Blockchain ledgers are stored in the efficient elastic storage files, satisfying the demand of fast storing massive amount of user data.
- Nodes of multiple roles and members can dynamically join or quit consortium blockchains.

## Cost-Effective

- Instance hibernation and waking at any time
- The Application Operations Management (AOM) service is used for comprehensive O&M on the BCS instances, providing system status, performance, and transaction monitoring, maintenance, and alarming to reduce O&M costs.
- The node scaling function greatly improves the cost-effectiveness.

## Secure and Private

Comprehensive approach to blockchain security:

- The security system ensures stable and secure running of blockchains.
- The Hyperledger-assured security system prevents data tampering and protects privacy by means of certificate management and the blockchain structure of data.
- Innovative algorithms such as homomorphic encryption and zero-knowledge proofs provide further privacy protection.
- OSCCA-published cryptographic algorithms are used for encryption and decryption.

## Trustworthy and Collaborative

BCS provides the Trusted Computing Platform to facilitate trusted cooperation between multiple parties. This platform has the following core features:

- Decentralized identity (DID) management, which is in compliance with the W3C DID and W3C verifiable credential (VC) standards. This feature lowers the threshold of trust and improves cooperation efficiency.
- Blockchain-based, trusted data sharing, which ensures trusted data flow between multiple parties, breaks data silos, and realizes data value.
- Confidential computing, which is based on blockchain, Trusted Execution Environment (TEE), and federated learning technologies. The raw data can be computed without being revealed, ensuring data privacy.

# 1.4 Key Concepts

## Blockchain

In a narrow sense, a blockchain is a list of data records (called blocks) linked in chronological order using cryptography and a distributed ledger to prevent data tampering and forging. In a broad sense, the blockchain technology is a new distributed infrastructure and computing paradigm that uses the blockchain data structure to verify and store data, distributed node consensus algorithms to generate and update data, cryptography to ensure security of data transmission and access, and smart contracts formed by automated scripts to implement programming and operate data.

## Distributed Ledger

A distributed ledger is a database shared, replicated, and synchronized among network members. It records transactions between network participants, such as exchange of assets and data. Use of a distributed ledger eliminates the time and expenditure of ledger reconciliation. Any reference to ledgers in BCS documents means distributed ledgers.

- Decentralized and trustless: Data copies are stored on nodes. No central node or a third-party organization is responsible for data control.
- Collectively maintaining data consistency: Each participant uses a public key as its identity. Nodes independently check the data validity and collectively determine the data to be written to the ledger, by consensus.
- Reliable data, difficult to be tampered with: Data is stored in blocks. Each node stores all blocks. Data access permissions can be customized. Block chaining prevents data tampering.

## Smart Contract

A smart contract, also called a chaincode, is a code logic that runs on a blockchain and is automatically executed under a specific condition. It is an important method for a user to implement service logic when using a blockchain. Thanks to the blockchain features, the execution results of smart contracts are reliable and cannot be forged or tampered with.

- Cheating is prevented. Smart contracts are automatically triggered when conditions are met. Execution results are verified independently.
- Results cannot be modified because the data is stored in the blockchain.
- Contract content is reliable because it is stored in the blockchain.
- Privacy is protected. Only specified participants can obtain contract content and data.

## Peer

Peers are network nodes that maintain ledgers. One or more peers form a peer organization.

## Orderer

Orderers are nodes that order transactions into a block.

## Channel

A channel isolates the ledger data of a transaction from that of other transactions 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.

## Distributed Consensus

A majority of independent participants in a system need to achieve consensus on a transaction or operation, for example, verification of double-spending

transactions, verification of service logic validity, and the decision on whether to write verified data to the existing ledger.

## Hash Algorithm

A hash value of a digital content segment can be used to verify data integrity. Any minor modification to digital content leads to a significant change in the hash value. A qualified hash algorithm can be used to easily obtain a hash value from digital content, but it is almost impossible to calculate the original digital content by using a hash value.

## Organization

A channel contains multiple members (organizations). If identity certificates of two entities on the blockchain network can be traced back to a same Root certificate authority (CA), the two entities belong to a same organization.

# 1.5 Edition Differences

## Enhanced Hyperledger Fabric

BCS provides basic and professional editions with different specifications. For details, see [Table 1-1](#). [Table 1-2](#) lists the cluster specifications.

### NOTE

Only one BCS instance can be deployed in a container cluster.

**Table 1-1** Comparison between editions

Item		Basic Edition	Professional Edition
Applicable scenario		Small scale commercial use	Medium-scale commercial use
Consortium blockchain		Supported	Supported
Peak transaction performance		≤ 500 TPS	≤ 2000 TPS
Consensus algorithm	Raft(CFT)	Supported	Supported
	FBFT	Not supported	Supported
Node management	Maximum number of organizations	2	5
	Maximum number of peers in an organization	2	2
	Maximum number of orderers	3	4

Item		Basic Edition	Professional Edition
	Maximum number of channels	2	4
	Automatic recovery from node faults	Supported	Supported
	Node auto scaling	Supported	Supported
Security functions	ECDSA	Supported	Supported
	OSCCA-published cryptographic algorithms	Not supported	Supported
High availability	Invoking smart contracts through RESTful APIs	Supported	Supported
	Common deployment	Supported	Supported
O&M and monitoring	O&M logging	Supported	Supported
	Node status monitoring	Supported	Supported
	Status alarms	Supported	Supported

**Table 1-2** Specifications

Edition	Cloud Container Engine (CCE) Cluster	Elastic Cloud Server (ECS)	Elastic IP (EIP) Address	VPCs and Subnets	Container Networking
Basic Edition	cce.s1.small (small-scale, single-master CCE cluster, supporting a maximum of 50 nodes) Single-AZ deployment	Specification: 4 vCPUs and 8 GB memory Quantity: Number of peers in the organization/ 2 + Number of orderers (1)	Private blockchain: no EIPs Consortium blockchain: one EIP for each cluster node EIP bandwidth: 1 Mbit/s	VPC: 1; subnet: 1	Tunnel network

Edition	Cloud Container Engine (CCE) Cluster	Elastic Cloud Server (ECS)	Elastic IP (EIP) Address	VPCs and Subnets	Container Networking
Professional Edition	cce.s2.small (small-scale, high-availability CCE cluster, supporting a maximum of 50 nodes) Multi-AZ deployment	Specification: 8 vCPUs and 16 GB memory Quantity: Number of peers in the organization/ 2 + Number of orderers	Private blockchain: no EIPs Consortium blockchain: one EIP for each cluster node EIP bandwidth: 5 Mbit/s	VPC: 1; subnet: 1	Tunnel network

## 1.6 Application Scenarios

### 1.6.1 Transactions Between Subsidiaries

BCS provides end-to-end (E2E) audit support for inter-subsidiary transactions by building a collaboration consortium with subsidiaries of a multinational company and audit organizations involved, developing trust and eliminating reconciliation and discrepancies between the transaction parties.

#### Industry Status Quo and Pain Points

- **Lack of trust between subsidiaries**  
Transaction parties do not fully trust in each other for ownership and fund transfer during contract execution and transactions.
- **Delayed financial settlement**  
Reconciliation of internal transactions can be extremely time and labor consuming. The discrepancy in reconciliation may lead to delayed settlement and report issuance.
- **Low efficiency and high cost**  
Internal reconciliation is time-consuming and requires a large number of financial personnel's efforts. However, the reconciliation result may still be incorrect, and it is hard to perform supervision.
- **No simple method of data sharing**  
The financial data of subsidiaries is distributed in different types of enterprise resource planning (ERP) systems, which are not integrated or connected.
- **Regulators lacking trust in company**  
A multinational company must keep data for many years (usually 10 or more years) and provide evidence to external auditors or authorities, demonstrating that data sources are trustworthy and the data has not been tampered with.



- **Restatements**

Inter-subsidiary transfer pricing and complex transactions may cause tax base erosion and profit shifting (BEPS) and may result in financial statement restatements.

## Solution Architecture

The BCS-based inter-subsidiary transaction solution has the following features:

- **Unified ledger**

Tamper-proof, consistent business transaction records are traceable, eliminating the necessity of reconciliation and meeting audit requirements.

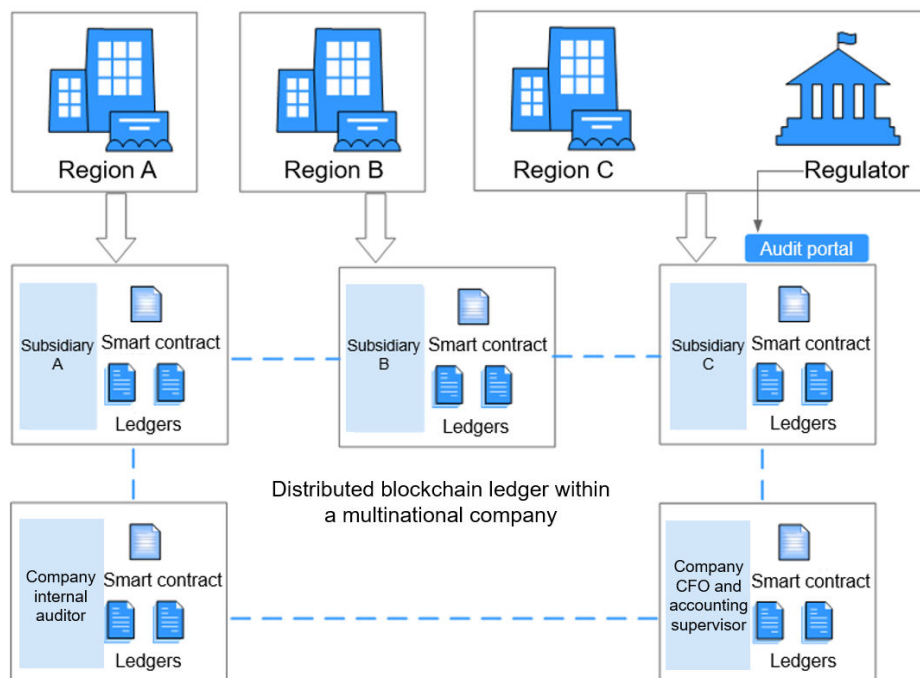
- **Digital assets**

Tokens are used to record the transaction assets and rights to realize the life-cycle management of digital assets.

- **Smart contract fulfillment**

Automated fulfillment ensures the fairness of transactions based on the contract terms and conditions.

**Figure 1-2** Solution architecture



## Solution Highlights

- Ensuring consistency of inter-subsidiary transaction records and the balance of accounting without the need for reconciliation
- Using tokens to follow goods' statuses, timing, locations, and ownership changes and strictly adhering to the contract clauses to carry out transactions, improving the trust between transaction parties
- Simplifying and normalizing the inter-subsidiary supply chain processes

- Supporting transactions that involve different systems
- Providing E2E traceable and immutable information for internal and external audits

## 1.6.2 Supply Chain Logistics

Manufacturers, warehousing institutes, logistics providers, and customers can use BCS to comprise collaboration consortia and use IoT technologies to record all the logistics information of goods, including production, warehousing, line haul transportation, reselling, and local logistics. The consortia break down information silos, improve circulation of information, and build trust between parties.

### Industry Status Quo and Pain Points

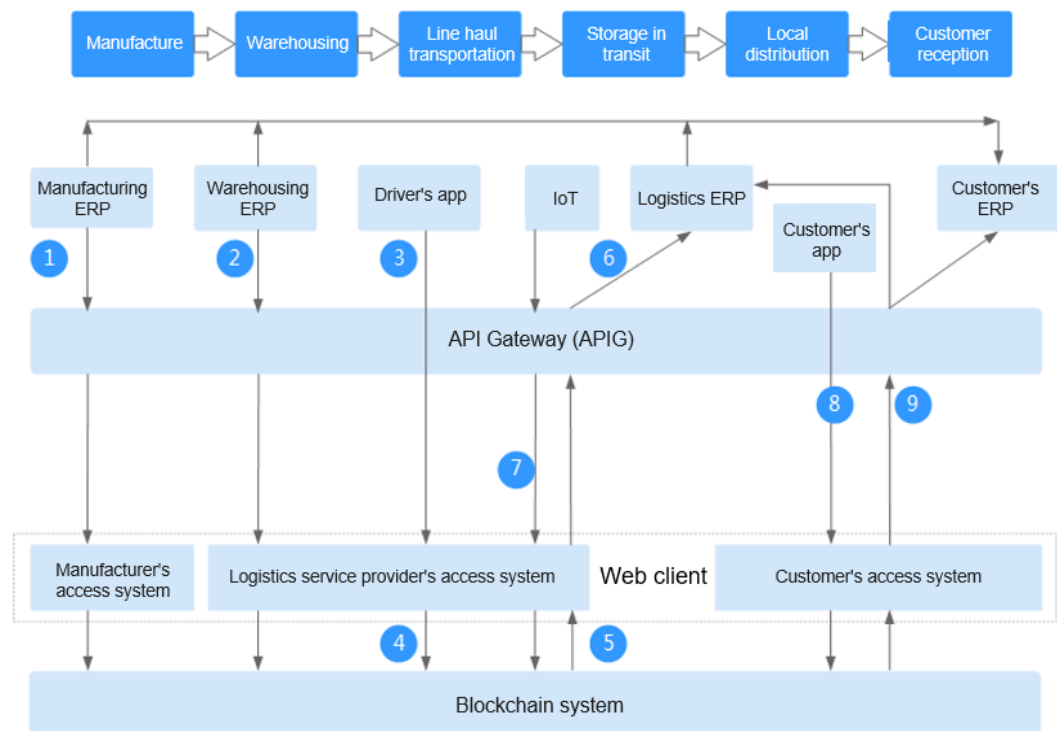
- **Disadvantage of using paper documents**  
Many phases of logistics still involve manual operations and paper documents. This causes long duration of the process, high costs, slow reconciliation, and risks of document losses or damage. The cost on maintaining and transferring documents accounts for 1/5 of the total logistics cost.
- **Low efficiency**  
Participants in a supply chain have their own information systems, independent from each other. There is no unified standard or system. It is difficult for them to collaborate effectively.
- **Long duration**  
Electronic information can be easily tampered with. Therefore, paper documents are used as the only type of proof for settlement, but extend the accounting period and the carriers' average collection period of receivables.
- **Difficult financing**  
Most carriers are small- and medium-sized enterprises, lacking credit records, scores, or credibility. Financing is difficult and requires high costs.

### Solution Architecture

The supply chain logistics solution provided by BCS can be combined with the IT information systems of logistic participants to achieve the following:

- Jointly maintain unified ledgers, which store immutable and traceable goods transfer records to meet audit requirements.
- Provide common APIs for participants' IT systems to access BCS and input data, which cannot be tampered with. In this way, participants establish their credibility and trust in each other.
- Automatically store the geo-fence information reported by the driver's app to show in real time when, where, and by whom goods are processed.
- Fulfill smart contracts to automatically perform signing, settlement, and calculation to obtain the performance data, which is considered fair due to the automation.

**Figure 1-3** Solution architecture



**Procedure:**

1. Goods delivery information is sent to the blockchain through the access system.
2. Goods reception and delivery information is sent to the blockchain through the access system.
3. The driver collects goods by scanning.
4. Logistics information is sent to the blockchain through the access system.
5. The blockchain system confirms that the received information has been stored.
6. The information is sent to the IT system through APIG.
7. The GPS data of trucks is sent to the blockchain through the access system.
8. The customer reception information is sent to the blockchain through the access system.
9. The blockchain system confirms that the received information has been stored and sent to the ERPs of the logistics service provider and manufacturer.

**Solution Highlights**

- **Reduced errors**  
Distributed, shared ledgers greatly improve the traceability and transparency of the supply chain and effectively reduce or eliminate changes of frauds and errors.
- **Increased efficiency**  
Electronic proofs of delivery (PODs) are used instead of paper documents to reduce the delay caused by paper works. Smart contracts enable automatic settlement to improve efficiency.
- **Lower costs**  
Quick settlement, automatic order reception, and goods follow-up significantly lower the logistics costs of all the involved parties.
- **Transparent audit**  
Immutability of distributed ledgers and non-repudiation of signatures allow for quick discovery of problems in supply chain logistics.
- **Trust**

In addition to transparent rules and automated settlement, the blockchain technology can help you follow goods all the way through production and transport to final reception. These mechanisms greatly improve the trust between all the involved parties.

### 1.6.3 Healthcare

BCS helps healthcare institutions, third-party organizations, and supervision departments to form a collaboration consortium. Healthcare information silos are broken down using electronic medical records that cannot be tampered with to protect privacy. This builds trust between doctors and patients and provides comprehensive health and medical care information for telemedicine and referral.

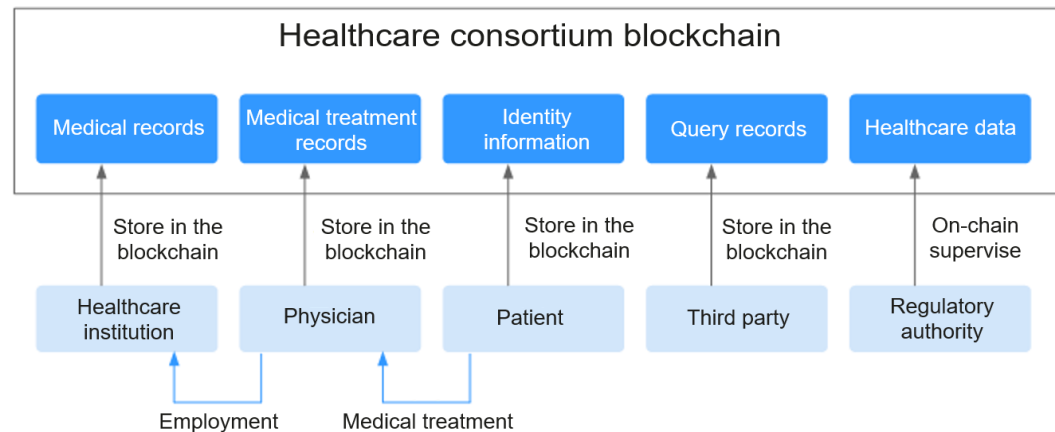
#### Industry Status Quo and Pain Points

- **Insecure data**  
Most healthcare data is stored in the data center. If a natural disaster or hacking occurs, patients' electronic medical records stored in the data center may be lost.
- **Information silos**  
There is no appropriate mechanism for mutual trust and data sharing between healthcare institutions, which leads to information silos and makes it difficult to obtain complete and comprehensive data. Data may be modified casually when shared and therefore, is considered unreliable.
- **Repeated medical treatment**  
Data is not shared between healthcare institutions. Performing repetitive health checks and creating new medical records are required when patients go to the different institutions, wasting time, money, and medical resources.
- **No access to personal medical data**  
Patients' medical data is stored in the hospital systems, however, patients cannot access to or manage it.

#### Solution Architecture

A healthcare consortium blockchain is built, comprising healthcare institutions, third parties, physicians, patients, and regulators based on electronic medical records (EMRs). The medical and healthcare data is stored in the blockchain and offered for queries or scientific research, with security and privacy protected by using encryption and smart contract-based authorization mechanisms.

**Figure 1-4** Solution architecture



## Solution Highlights

- Information silos broken down**  
 The healthcare consortium blockchain connects information systems of healthcare institutions, so that regional inspection as well as ultrasound and radiological examination results can be securely exchanged for online healthcare, two-way referral, and remote consultation.
- Immutable medical data**  
 The EMRs, physicians' diagnosis process and results, medical record query histories, and patient identity information are transparently stored in blockchains to ensure that they cannot be tampered with. This reduces medical disputes and constructs a harmonious healthcare environment.
- Protected privacy and right to know**  
 Encryption and smart contract-based authorization mechanisms offer patients access to their own healthcare data while protecting their privacy. Others can access the data only when authorized.
- Quick and effective supervision**  
 Regulatory authorities can use the data on blockchains to effectively prevent healthcare treatment that violates regulations, reducing medical disputes.

## 1.7 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your BCS resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your resources.

With IAM, you can use your account to create IAM users, and assign permissions to the users to control their access to specific resources. For example, some software developers in your enterprise need to use BCS resources but should not be allowed to delete the resources or perform any other high-risk operations. In this scenario, you can create IAM users for the software developers and grant them only the permissions required for using BCS resources.

If your account does not require individual IAM users for permissions management, skip this section.

## Enhanced Hyperledger Fabric

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

BCS is a project-level service deployed and accessed in specific physical regions. To assign BCS permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. Switch to a region where you have been authorized to access BCS.

**Table 1-3** lists the system-defined policy supported by enhanced Hyperledger Fabric.

**Table 1-3** System-defined roles and policies supported by enhanced Hyperledger Fabric

Role/Policy Name	Description	Type	Dependency
BCS Administrator	BCS administrator	System-defined role	Tenant Guest, Server Administrator, ELB Administrator, CCE Cluster Admin, SFS Administrator, SWR Admin, APM FullAccess, AOM FullAccess, CCE Administrator, VPC Administrator, EVS Administrator, ECS FullAccess, DMS Administrator, and BSS Administrator
BCS FullAccess	Full permissions for BCS	System-defined policy	None
BCS ReadOnlyAccess	Read-only permissions for BCS	System-defined policy	None

### NOTE

If you select BCS FullAccess, you need to set namespace permissions for the CCE cluster because BCS depends on CCE and the CCE namespace uses Kubernetes RBAC authorization. See [Permissions Management](#) to set namespace permissions. Suggestion: Create a cluster in CCE, grant the **cluster-admin** permission to the user groups/users in the cluster. Then, go to the BCS console, create a BCS instance. On the **Configure Resources** page, select **Use an existing CCE cluster** for **Cluster**.

## 1.8 Billing

BCS provides basic and professional editions, with different specifications and fees.

### Billing Items

Table 1-4 Billing Item

Billing Mode	Item 1	Item 2	Billing Formula
Pay-per-use	BCS instance	Peer	Price = Fees of BCS instances + Fees of peers (2 peers for free)

For details, see product pricing details.

### Billing Mode

- Pay-per-use: BCS instances are charged by actual duration of use, with a billing cycle of one hour.

 **NOTE**

The above-mentioned prices are for BCS instances only. Resources used for BCS, such as ECS nodes, CCE instances, EVS disks, EIPs, and bandwidth resources are billed separately.

- Billing formula: Price = Fees of BCS instances + Fees of peers (2 peers for free)  
(Note: Each instance has 2 free peers and you only pay for the extra peers.)

## 1.9 Restrictions

To use BCS, you must create CCE clusters, bind EIPs to servers, deploy a BCS instance, and build a blockchain application.

A maximum of five enhanced Hyperledger Fabric instances can be created. The specifications of each instance vary depending on the edition. For details, see [Edition Differences](#).

# 2 Managing Enhanced Hyperledger Fabric Instances

---

## 2.1 BCS Overview

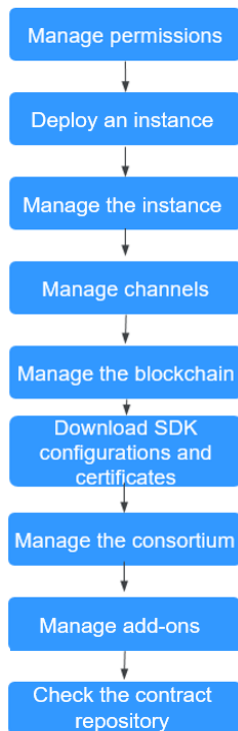
Blockchain Service (BCS) allows you to deploy instances, and manage blockchains, channels, members, and notifications. The following figure outlines the BCS usage process.

 **NOTE**

BCS does not involve sensitive user information. Which, why, when, and how data is processed by BCS must comply with local laws and regulations. If sensitive data needs to be transmitted or stored, encrypt data before transmission or storage.



**Figure 2-1** Outline of the BCS usage process



1. **Manage permissions.**  
Create a user and grant BCS permissions.
2. **Deploy an instance.**  
Enhanced Hyperledger Fabric instances can be deployed in CCE clusters.
3. **Manage the instance.**  
You can view the running statuses of your enhanced Hyperledger Fabric instances and perform operations on them.
4. **Manage channels.**  
Peers communicate through channels. You can create channels and add organizations and peers to them.
5. **Manage the blockchain.**  
You can manage chaincodes on the web, including installing, instantiating, and updating chaincodes.
6. **Download SDK configurations and certificates.**  
Before developing an application, download the configuration file which contains the user certificate and SDK.
7. **Manage the consortium.**  
After creating a consortium blockchain, you can invite tenants to join it.
8. **Manage add-ons.**  
Add-ons allow you to extend the functionality of BCS instances as required.
9. **Check the contract repository.**  
The contract repository provides smart contract templates that can implement certain functions. You can directly use the code provided by the templates or use the templates as a foundation for developing your own smart contracts.

## 2.2 Permissions Management

### 2.2.1 Creating a User and Granting BCS Permissions

This section describes how to use **IAM** to implement fine-grained permissions control for your BCS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing BCS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account or a cloud service to perform professional and efficient O&M on your BCS resources.

If your account does not require individual IAM users, skip this chapter.

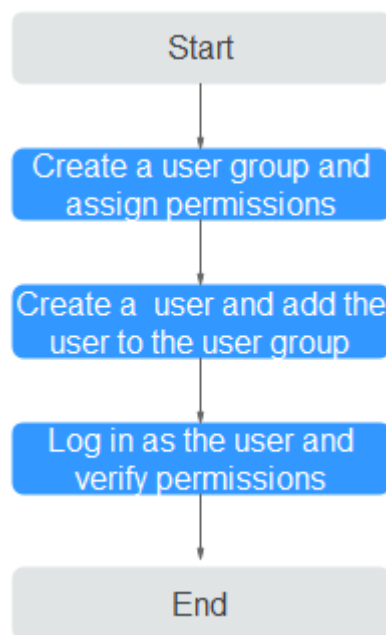
This section describes the procedure for granting permissions (see **Figure 2-2**).

#### Prerequisites

Learn about the permissions (see **Cluster Permissions (IAM-based)**) supported by BCS and choose policies or roles according to your requirements.

#### Process Flow

**Figure 2-2** Process of granting BCS permissions



1. **Create a user group and assign permissions to it.**

- Create a user group on the IAM console, and assign the BCS Administrator policy to the group.
- 2. **Create a user and add the user to the user group.**

Create a user on the IAM console and add the user to the group created in 1.
- 3. **Log in** and verify permissions.
 

Log in to the BCS console as the created user, and verify that the user has the BCS operating permissions.

## 2.2.2 Creating a Custom Policy

Custom policies can be created to supplement the system-defined policies of BCS.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit policies from scratch or based on an existing policy in JSON format.

For details, see [Creating a Custom Policy](#). The following section contains examples of common BCS custom policies.

**Step 1** On the management console homepage, click **Identity and Access Management**.

**Step 2** In the navigation pane, choose **Permissions > Policies/Roles** and click **Create Custom Policy**.

**Step 3** On the **Create Custom Policy** page, set the policy name, view, content, and description, then click **OK**.

- **Policy Name:** Enter a custom policy name, for example, "partial BCS permissions".
- **Policy View:** Select **JSON**.
- **Policy Content:** Enter the policy content based on the template.

For example, copy the following content to grant permissions for instance, channel, and member management.

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "bcs:blockchains:list",
        "bcs:channels:list",
        "bcs:members:list"
      ]
    }
  ]
}
```

**Table 2-1** Policy content parameters

Parameter	Description	Setting
Version	Policy version	Fixed to <b>1.1</b> .

Parameter		Description	Setting
Statement	Effect	Whether the actions are allowed	<ul style="list-style-type: none"> <li>- <b>Allow</b></li> <li>- <b>Deny</b></li> </ul>
	Action	Operations to be performed on BCS	Each action name is in the format of <i>Service name:Resource type:Operation</i> and cannot be customized. <a href="#">Table 2-2</a> lists the fine-grained permissions supported by BCS. After you set any action, the permissions for the action will be granted to the IAM user.

**Table 2-2** Action description

Related Action	Action Description
bcs:peer:get	Querying peers
bcs:notifications:list	Managing notifications
bcs:blockchain:get	Querying BCS service details
bcs:notification:get	Querying notification details
bcs:membertopo:get	Querying topology information
bcs:contract:get	Viewing contract details
bcs:member:get	Querying member details
bcs:plugin:get	Querying add-on details
bcs:dashboard:get	Viewing the dashboard
bcs:sdkcfg:post	Downloading SDK configurations
bcs:blockchainondemand:create	Creating BCS services
bcs:blockchain:awakehibernate	Freezing or unfreezing instances
bcs:notification:put	Processing notifications
bcs:eip:put	Updating EIPs
bcs:plugin:delete	Deleting add-ons
bcs:taskserver:create	Creating the taskserver add-on
bcs:member:put	Inviting tenants
bcs:notification:delete	Deleting notifications

Related Action	Action Description
bcs:channel:create	Creating channels
bcs:member:delete	Deleting members
bcs:channel:put	Adding peers to channels
bcs:blockchain:upgrade	Upgrading or rolling back BCS services
bcs:cert:post	Downloading certificates
bcs:blockchain:delete	Deleting BCS services
bcs:channel:delete	Deleting a channel
bcs:members:list	Listing members
bcs:channels:list	Listing channels
bcs:plugins:list	Listing add-ons
bcs:blockchains:list	Listing BCS services
bcs:contracts:list	Listing contracts
bcs:restapi:create	Creating the baas-restapi add-on
bcs:cluster:post	Cluster-related operations

----End

## 2.3 Instance Deployment

### 2.3.1 Deployment Using a CCE Cluster

BCS instances can be deployed using Cloud Container Engine (CCE). CCE is a high-performance, high-reliability service through which enterprises can manage containerized applications. CCE supports native Kubernetes applications and tools, allowing you to easily set up a container runtime environment on the cloud. For more information, see *Cloud Container Engine User Guide*. When creating a BCS instance, configure basic parameters and network nodes to quickly create and deploy the instance.

Enhanced Hyperledger Fabric instances can be deployed in CCE clusters. This section describes how to deploy an enhanced Hyperledger Fabric instance using a CCE cluster.

 NOTE

- The BCS instance will use the CCE cluster exclusively. Ensure that the CCE cluster is available before you deploy the BCS instance.
- You can prepare a CCE cluster in advance, and select it when you create an enhanced Hyperledger Fabric instance. Alternatively, you can customize a CCE cluster or select **Quick Config** to use the default specifications when you create an enhanced Hyperledger Fabric instance.

## Prerequisites

Only IAM users with robust permissions can create BCS instances. For details, see [Permissions Management](#).

You can create a user group, grant permissions to the user group, and then add the user to the user group. In this way, the user has the permissions of the user group.

## Deploying a BCS Instance

After the environment is ready, perform the following steps to create a BCS instance:

**Step 1** Create a blockchain instance.

Log in to the BCS console, click **Instance Management**, and click **Create BCS Instance** in the upper right corner.

**Step 2** Configure basic information about the BCS instance by referring to [Table 2-3](#).

**Table 2-3** Basic information parameters

Parameter	Description	Example Setting
Region	Select the region where the blockchain infrastructure is located. You are advised to select the same region as the service application system.	Retain the default value.
Enterprise Project	Select an existing enterprise project, to which the BCS instance will be added. <b>NOTE</b> <ul style="list-style-type: none"><li>• If the Enterprise Management service is not enabled, this parameter is unavailable.</li><li>• When deploying an instance in an existing CCE cluster, choose the same enterprise project as that used by the cluster to ensure instance performance.</li></ul>	default

Parameter	Description	Example Setting
Instance Name	An instance name can contain 4 to 24 characters, including letters, digits, and hyphens (-). It cannot start with a hyphen (-). <b>NOTE</b> Currently, the name of a created BCS instance cannot be changed. You can only create a new instance with a new name.	Enter <b>bcs-wh</b> .
Edition	BCS provides basic and professional editions. <b>NOTE</b> Editions cannot be changed for a deployed BCS instance.	Select <b>Professional</b> .
Blockchain Type	A private blockchain is used only by the tenant that creates it. A consortium blockchain can be used by multiple tenants.	Select <b>Private</b> .
Enhanced Hyperledger Fabric Version	BCS instance version. BCS v4.x.x corresponds to Hyperledger Fabric v2.2.	Select <b>v2.2</b> .
Consensus Mechanism	The supported mechanisms for blockchain nodes reaching consensus include: Raft (crash fault tolerant) and Fast Byzantine fault tolerance (FBFT). <b>NOTE</b> If Raft (CFT) is selected, a basic or professional edition instance has three orderers by default.	Select <b>FBFT</b> .
Resource Access Initial Password	Password of blockchain administration user <b>admin</b> , ECS user <b>root</b> , or CouchDB database user. It will be used as such a password if you do not set <b>Blockchain Mgmt. Initial Password, Password of Root User, or Initial Password</b> displayed when <b>NoSQL (CouchDB)</b> is selected for <b>Ledger Storage</b> .	-
Confirm Password	Confirm the resource access initial password.	-

**Step 3** (Optional) Click **Quick Config** to allow the system to automatically create an instance with the specifications listed in [Table 2-4](#).

**Table 2-4** Default specifications

Item	Basic Edition	Professional Edition
Number of CCE cluster nodes	1	2
CCE node specifications	4 vCPUs   8 GB	4 vCPUs   8 GB
	Note: If the default specifications cannot be selected, other higher specifications will be created by default.	
Storage space of SFS Turbo	510 GB	510 GB
EIP	Type: Dynamic BGP; Bandwidth: 5 Mbit/s	

**Step 4** Click **Next: Configure Resources**. [Table 2-5](#) describes the resource parameters.

**Table 2-5** Resource parameters

Parameter	Description	Example Setting
Environment Resources	Use the default environment or customize your environment resources.	Select <b>Custom</b> .
Cluster	Cluster where the BCS instance will be deployed. You can use an existing cluster or create a new CCE cluster. <b>NOTE</b> CCE clusters of v1.21 or earlier are supported.	Select <b>Create a new CCE cluster</b> .
AZ	Select the AZ where the ECS is located.	Select <b>AZ1</b> .
ECS Specifications	Specifications of the ECSs in the CCE cluster.	Select the flavor for <b>4 vCPUs   8 GB</b> .
ECS Quantity	Enter the required ECS quantity.	Enter <b>2</b> .
High Availability	If you have high requirements on system reliability, create high-availability ECSs.	Yes
VPC	You can create a new virtual private cloud (VPC), select an existing VPC, or let the system automatically create a VPC.	Select <b>Automatically create VPC</b> .
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security.	Select <b>Automatically create subnet</b> .
ECS Login Method	Either a password or key pair can be used to log in to ECSs.	Select <b>Password</b> .



Parameter	Description	Example Setting
Password of Root User	Password of the root user for logging in to ECSs. If you do not enter a password here, the previously specified resource access initial password will be used.	-
Confirm Password	Confirm the ECS login password of the root user.	-
Use EIP of a CCE Node	<ul style="list-style-type: none"> <li>If you select <b>Yes</b>, an EIP bound to the cluster will be used as the blockchain network access address. If the cluster is not bound with any EIP, bind an EIP to the cluster first.</li> <li>If you select <b>No</b>, a private address of the cluster will be used as the blockchain network access address. Ensure that the application can communicate with the internal network of the cluster.</li> </ul>	Select <b>Yes</b> .
EIP Billed By	Specifies whether the bandwidth is charged by fixed bandwidth or by traffic.	Select <b>Bandwidth</b> .
EIP Bandwidth	Select a bandwidth as required.	Set it to 5 Mbit/s.

**Step 5** Click **Next: Configure Blockchain**. [Table 2-6](#) describes the blockchain parameters.

**Table 2-6** Blockchain parameters

Parameter	Description	Example Setting
Blockchain Configuration	Use the default blockchain configurations or customize your own blockchain configurations.	Select <b>Custom</b> .
Blockchain Mgmt. Initial Password	Enter the blockchain management initial password. If you do not enter a password here, the previously specified resource access initial password will be used.	-
Confirm Password	Enter the blockchain management initial password again for confirmation.	-
Volume Type	<b>SFS Turbo</b> provides low-latency and high-IOPS file storage.	Select <b>SFS Turbo</b> .

Parameter	Description	Example Setting
Storage Capacity of Peer Organization (GB)	Stores shared distributed ledger, consensus data, and other intermediate data of the blockchain system.	Set it to 500 GB.
Ledger Storage	File database (GoLevelDB) and NoSQL (CouchDB) are supported. <ul style="list-style-type: none"> <li>File database (GoLevelDB): The Fabric native storage mode is used. Historical transaction data is stored in the blockchain, and status data is stored in the LevelDB.</li> <li>NoSQL (CouchDB): The CouchDB storage mode supported by the Fabric is used to store transaction data and status data. Each CouchDB database is a collection of independent documents. Each document maintains its own data and self-contained schema.</li> </ul>	Select <b>File database (GoLevelDB)</b> .
Peer Organization	Peer organizations to be added to the BCS instance. <ul style="list-style-type: none"> <li>If you use an existing cluster, customize the peer organization name and peer quantity. <b>Automatically create SFS Turbo file system</b> will be displayed in the <b>Network Storage</b> area.</li> <li>If you use a new CCE cluster, customize the peer organization name and peer quantity.</li> </ul>	Add a peer organization named <b>organization</b> with 2 peers.
Channel Configuration	Channels isolate business in a consortium blockchain. Business participants (some or all of the organizations in a consortium) are channel members. Each channel can be regarded as a sub-chain and corresponds to one distributed ledger.	By default, a channel named <b>channel</b> has been created, and the peer organization you just specified has been added to the channel.
Orderer Quantity	Number of nodes that order transactions into blocks in the blockchain network. When the consensus mechanism is Raft (CFT), the number of orderers is 3.	Enter <b>3</b> .

Parameter	Description	Example Setting
Security Mechanism	Encryption algorithm used to ensure data security. ECDSA and OSCCA-published cryptographic algorithms are supported.	Select <b>ECDSA</b> .
Configure Block Generation	The configuration of block generation includes the block generation interval, maximum number of transactions in a block, and maximum size of a block. A new block is generated at the specified interval or when the transaction quantity or size of a block reaches the threshold. Configure these parameters based on the transaction frequency and service volume.  Select <b>Yes</b> or <b>No</b> as required. <ul style="list-style-type: none"> <li><b>Yes:</b> Set the block generation interval, transaction quantity per block, and block size as required.</li> <li><b>No:</b> You do not need to set parameters. By default, the block generation interval is 2 seconds, the number of transactions per block is 500, and the block size is 2 MB.</li> </ul>	Select <b>No</b> .
Enable Support for RESTful API	If you need to use RESTful APIs to invoke chaincodes, select <b>Yes</b> .	Select <b>No</b> .

**Step 6** Click **Next: Confirm**.

**Step 7** Confirm the configurations and click **Submit**.

Wait for several minutes. After a message is displayed indicating successful installation, check the status of the instance. If it is **Normal**, the deployment is completed.



----End

## Subsequent Operations (Optional)

View the operation records of creating, deleting, and upgrading instances, adding organizations, expanding peers, creating channels, and adding peers to channels. In the left part of the window, you can filter records by status, including **In progress**, **Upgrading**, **Deleting**, **Finished**, and **Failed**. The figure is for reference only.

**Figure 2-3** Operation records

Task Details

	Resource Name	Resour...	Operat...	Operation ...	Cluster	Created	Operation
▼	bcs-c14mku	BCS In...	Create	Failed	cluster-bc...	Dec 19, 2022 14:12:59 GMT...	<a href="#">View Details</a> <a href="#">Delete</a>
▼	bcs-p6up8c	BCS In...	Create	Successful	bcs-nodel...	Dec 19, 2022 11:47:46 GMT...	<a href="#">View Details</a> <a href="#">Delete</a>
▼	bcs-et7lj	BCS In...	Create	Failed	bcs-nodel...	Dec 19, 2022 09:31:59 GMT...	<a href="#">View Details</a> <a href="#">Delete</a>

The system stores records of the latest three days.

**Step 1** Log in to the BCS console. In the navigation pane, click **Instance Management**.

**Step 2** Click **Task Details**.

Search records by the resource name. You can also view details or delete records.

----End

You can configure an anti-affinity label for the cluster node where the BCS instance is deployed. This label can be used to isolate the instance from other applications in the same cluster to ensure normal running of the system.

**Step 1** Log in to the CCE console. In the navigation pane, choose **Resource Management** > **Nodes**. The node list is displayed. Choose **Operation** > **Manage Label** in the **Operation** column.

**Step 2** Click **Add Label**. Set **Key** to **nodeScope** and **Value** to **userApplication** for the label to be added.

**Step 3** Click **OK**. After **Label updated successfully** is displayed, click **Manage Labels** again. Then you can see the label that you have added.

----End

## 2.4 Instance Management

### 2.4.1 Basic Operations

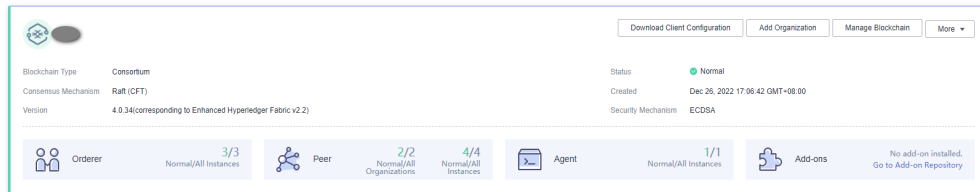
You can view the running statuses of your enhanced Hyperledger Fabric instances and perform operations on them.

#### Procedure

**Step 1** Log in to the BCS console.

**Step 2** In the navigation pane, click **Instance Management**. You can view the overall running status of your instances. For details about the parameters, see [Table 2-7](#).

**Figure 2-4** Viewing an enhanced Hyperledger Fabric instance



**Table 2-7** Parameters

Parameter	Description
Blockchain Type	Type of the blockchain, that is, <b>Consortium</b> or <b>Private</b> .
Consensus Mechanism	Consensus mechanism used by the instance, for example, <b>Raft (CFT)</b> . The following consensus mechanisms are supported: <ul style="list-style-type: none"> <li>• <b>FBFT</b>: The fast Byzantine fault tolerance (FBFT) algorithm. It requires 4 to 10 orderers for transaction ordering and tolerates faults at a maximum of <math>(N - 1)/3</math> orderers, where N indicates the total number of orderers. It supports Fabric v2.2.</li> <li>• <b>Raft (CFT)</b>: A CFT ordering instance that tolerates faults at a maximum of <math>(N - 1)/2</math> orderers, where N indicates the total number of orderers. It supports Fabric v2.2.</li> </ul>
Version	BCS instance version.
Status	Status of the BCS instance, which can be <b>Unknown, Normal, Abnormal, Creating, Upgrading, Adding peers, EIP abnormal, Deleting, Hibernating, or Cluster frozen</b> .
Created	Time when the BCS instance was created, for example, <b>Dec 10, 2022 20:30:21 GMT+08:00</b> .
Security Mechanism	Encryption algorithm used to ensure data security.
Orderer	Numbers of normal and abnormal orderer organizations.
Peer	Numbers of peer organizations and instances.
Agent Peer	Numbers of normal and abnormal agent organizations.
Add-ons	Number of add-ons. For example, <b>1/2</b> indicates that the total number of instances is 2 and 1 instance is normal.

**Step 3** On the **Instance Management** page, you can perform operations listed in [Table 2-8](#).


**Table 2-8** Operations

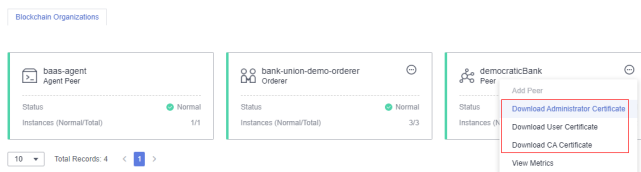
Category	Operation	Description
Organization management	Adding an organization	<ol style="list-style-type: none"> <li>On an instance card, click <b>Add Organization</b>, and specify the organization name, peer quantity, and network storage instance.</li> <li>Click <b>Next</b>.</li> </ol> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>Do not perform operations on the instance when adding an organization.</li> <li>After you add an organization to an existing channel, update the endorsement policy of the channel before instantiating the chaincode. Otherwise, the instantiation may fail due to a certificate verification failure.</li> </ul>
Instance management	Downloading client configurations	Before developing an application, download the SDK configurations and application certificates for accessing the blockchain network. On the <b>Instance Management</b> page, click <b>Download Client Configuration</b> and select configuration files to download, including the SDK configuration file, orderer certificate, and peer certificates. For details, see <a href="#">Downloading SDK Configurations and Certificates</a> .
	Managing the blockchain	This operation is available only after an EIP is bound. On an instance card, click <b>Manage Blockchain</b> to view, install, instantiate, upgrade, and delete chaincodes.
	Upgrading the version	<p>A BCS instance can be upgraded to the latest version if <b>Upgradable</b> is displayed in the upper left corner of the instance card. The operations are as follows:</p> <ol style="list-style-type: none"> <li>Log in to the BCS console.</li> <li>In the navigation pane, click <b>Instance Management</b>.</li> <li>Choose <b>More &gt; Upgrade</b> on an instance card.</li> <li>View the current instance version or upgrade the BCS instance to the latest version.</li> </ol> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>Before upgrading your consortium blockchain instance, reach an agreement with other members to eliminate effects on their instances.</li> <li>Do not initiate version upgrade when the chaincode is being installed or instantiated.</li> <li>You can only upgrade an instance from an earlier version to a later version. Rollback is supported only if the upgrade fails.</li> </ul>

Category	Operation	Description
	Rolling back upgrade	<p>If the version fails to be updated, you can roll back the upgrade. The operations are as follows:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console.</li> <li>2. In the navigation pane, click <b>Instance Management</b>.</li> <li>3. Choose <b>More &gt; Roll Back Version</b> on an instance card.</li> <li>4. During the rollback, the instance status is <b>Upgrading</b>. After the rollback is completed, the instance status is <b>Normal</b>.</li> </ol> <p><b>NOTE</b> Instances failed the upgrade can be upgraded again after the rollback.</p>
	Resetting the management password	<p>Choose <b>More &gt; Reset Management Password</b> on an instance card. By default, resetting this password will also reset the password for logging in to the Blockchain Management console. If you do not want to reset the password, change the passwords on the Blockchain Management console.</p>
	Changing the blockchain network access address	<p>Choose <b>More &gt; Change Access Address</b> on an instance card, select a new address, and click <b>OK</b>.</p>
	Hibernating	<p>Choose <b>More &gt; Hibernate</b> on an instance card, and click <b>OK</b>.</p> <p><b>NOTE</b> Only instances in the <b>Normal</b> state can be hibernated.</p>
	Waking	<p>Choose <b>More &gt; Wake</b>, and click <b>OK</b>.</p> <p><b>NOTE</b> Only instances in hibernation can be woken.</p>
	Deleting	<p>Choose <b>More &gt; Delete</b>.</p>


**Step 4** Click an instance name to view the instance details.

- Viewing instance basic information  
On the **Basic Information** tab page, view the instance details, agent peers, orderers, peers, CPU usage, and physical memory usage.
- Monitoring data  
On the **Monitoring** tab page, view monitoring data about the instances.  
For details about how to view monitoring information, see [Viewing Monitoring Data and Logs](#).

- Viewing logs  
On the **Logs** tab, view the logs of the organization instances and add-on instances.  
For details about how to view log information, see [Viewing Monitoring Data and Logs](#).
- Downloading certificates  
In the **Blockchain Organizations** area on the **Basic Information** tab page, click  to download the certificates.

**Figure 2-5** Downloading certificates**NOTE**

You can click **Download Client Configuration** on an instance card to download the SDK and certificates. For details, see [Downloading SDK Configurations and Certificates](#).

- Adding peers  
In the **Blockchain Organizations** area on the **Basic Information** tab page, click , and click **Add Peer**. Specify the peer quantity, confirm the configurations, and click **Submit**.

**NOTE**

- Do not perform operations on the instance when adding peers.
- Each organization supports a maximum of 2 peers in a basic or professional edition instance. No more peers can be added after the number of peers has reached the maximum allowed limit.

----End

## 2.4.2 O&M Center

### 2.4.2.1 Viewing Monitoring Data and Logs

BCS provides O&M monitoring capabilities. Technical support engineers can view the monitoring data and logs on the BCS console.

#### Viewing Monitoring Data

- Step 1** Log in to the BCS console.
- Step 2** In the navigation pane, click **Instance Management** to view the basic information of a BCS instance, including the blockchain type, consensus mechanism, status, and creation time.



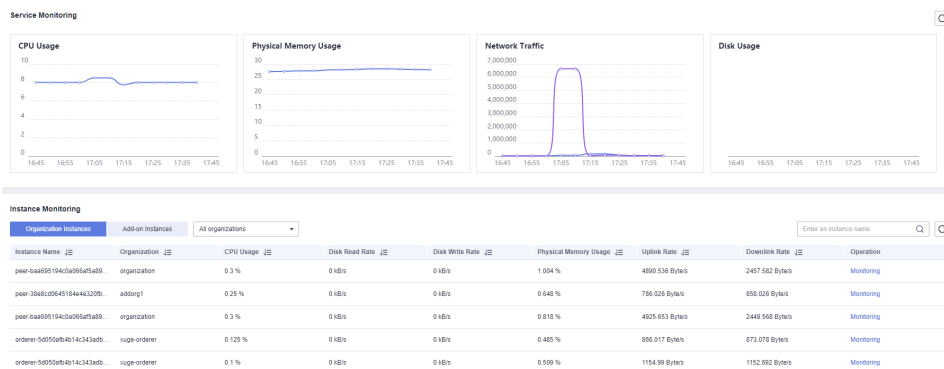
**Step 3** On an instance card, click the instance name.

**Step 4** Click the **Monitoring** tab to view the service monitoring and instance monitoring data.

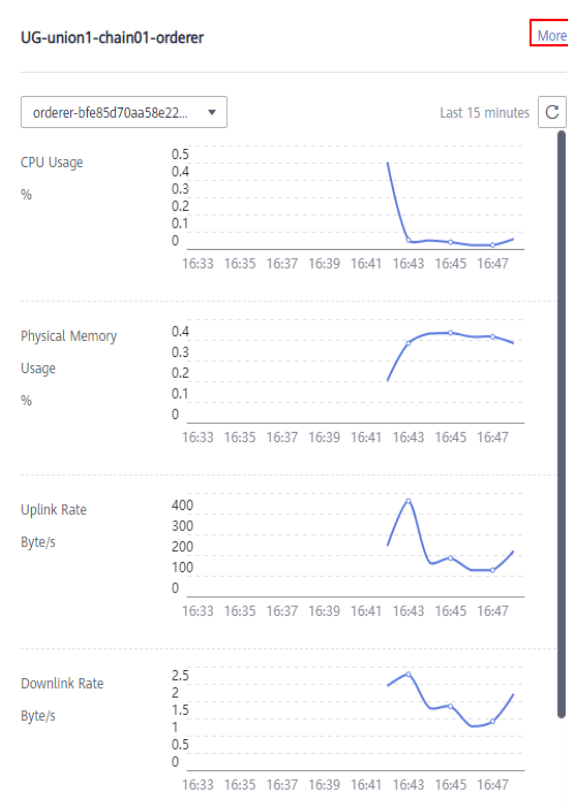
- Service monitoring allows you to view the CPU usage, physical memory usage, network traffic, and disk usage of the service.
- Instance monitoring allows you to view the organization instance information, including the CPU usage, disk read rate, disk write rate, physical memory usage, uplink rate, and downlink rate.

You can click **View Metrics** to view the data of the last 15 minutes. You can also click **More** to view more monitoring data.

**Figure 2-6** Viewing monitoring information



**Figure 2-7** Viewing more monitoring data



----End

## Viewing Logs

**Step 1** Log in to the BCS console.

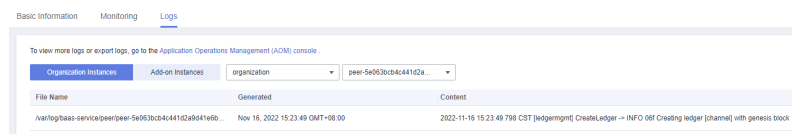
**Step 2** In the navigation pane, click **Instance Management** to view the basic information of a BCS instance, including the blockchain type, consensus mechanism, status, and creation time.

**Step 3** On an instance card, click the instance name.

**Step 4** Click the **Logs** tab. By default, log data in the last 5 minutes is displayed, including the log file name, creation time, and log content.

To view more logs or export logs, go to the AOM console.

**Figure 2-8** Viewing logs



----End

### 2.4.2.2 Viewing Alarms


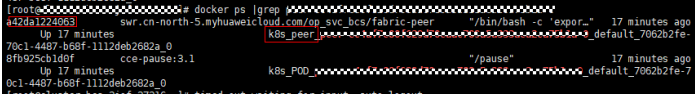
BCS provides O&M monitoring capabilities. Technical support can view alarms generated in BCS and CCE. [Table 2-9](#) lists common alarms.

#### NOTE


Perform preliminary checks based on the following table. If the alarm persists, contact technical support.

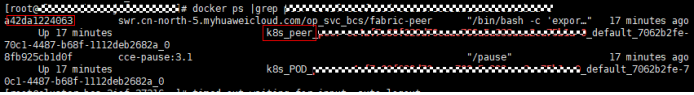
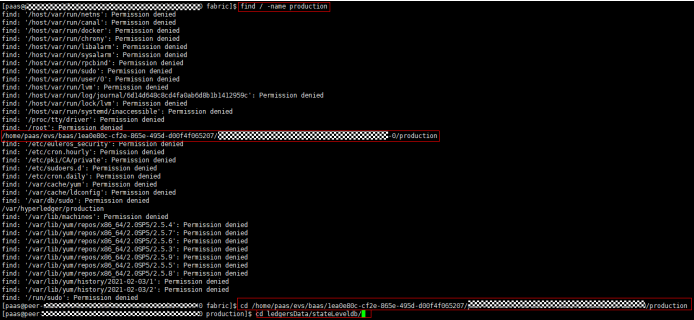
If an alarm is generated in CCE, and BCS instances are running properly, refer to .

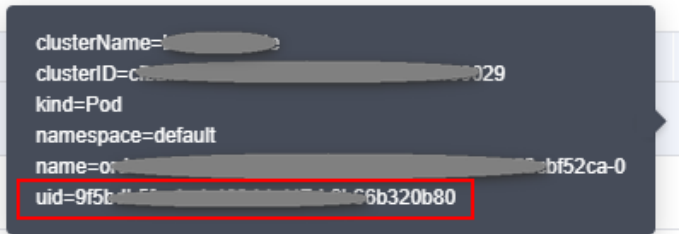
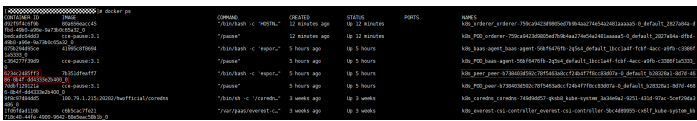
**Table 2-9** Common alarms for BCS

Alarm Name	Alarm Source	Solution
PeerConnect Failed	BCS	<p>Peers fail to connect to orderers. Possible causes include:</p> <ul style="list-style-type: none"> <li>• The network may have fluctuated.</li> <li>• The orderer is abnormal.</li> </ul> <p>If the network fluctuates, the alarm will be automatically cleared within a few minutes.</p> <p>If the alarm persists and is not cleared after a few minutes, the peer may have been disconnected from the orderer. In this case, perform the following steps:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console, click <b>Instance Management</b> in the navigation pane, and click an instance to go to the instance details page.</li> <li>2. On the BCS instance details page, click the <b>Monitoring</b> tab and then the <b>Active</b> tab. Record the value of <b>name</b> in the <b>Resource Name</b> column.</li> </ol> <p><b>Figure 2-9</b> Checking name of the failed peer</p>  <pre> clusterName=bcscn-north-5-myhuaweicloud.com clusterID=cfb2...00029 kind=Pod namespace=default name=prd...52ca-0 uid=9f...320b80     </pre> <ol style="list-style-type: none"> <li>3. Log in to all nodes (bound with EIPs) in the CCE cluster where the instance is deployed and run the <b>docker ps   grep name</b> command (as shown in the following figure). The container whose name starts with <b>k8s_peer</b> (or <b>k8s_orderer</b> for an orderer) is the container for which the alarm is generated. The container ID is at the start of the section.</li> </ol> <p><b>Figure 2-10</b> Viewing the command output</p>  <pre> [root@4526a122466d]# docker ps   grep k8s_peer Up 17 minutes 70c1-4487-b68f-1112deb2682a_0 k8s_peer_... 8fb925cb1d0f cce-pause:3.1 /pause" 17 minutes ago Up 17 minutes 9c1-4487-b68f-1112deb2682a_0 k8s_POD_..._default_7062b2fe-7     </pre> <p><b>NOTE</b> For details about how to log in to a node in a CCE cluster, see <a href="#">Viewing O&amp;M Logs on a Backend VM</a>.</p> <ol style="list-style-type: none"> <li>4. Check whether the container is normal.</li> <li>5. If the container is abnormal, run the <b>docker restart Container ID</b> command to restart the container.</li> </ol>

Alarm Name	Alarm Source	Solution
		6. If the fault persists, go to <b>Log &gt; Log Files</b> on the AOM console. Download the log files of the peer and orderer on the cluster for which the alarm is generated, and send the log files to technical support.

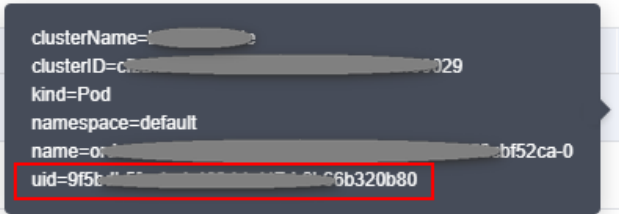

Alarm Name	Alarm Source	Solution
PeerWriteDB Failed	BCS	<p>A peer fails to access database files. Possible causes include:</p> <ul style="list-style-type: none"> <li>The status database file is damaged or lost.</li> <li>The storage service mounted to the status database is deleted.</li> </ul> <p>To rectify this fault, perform the following steps:</p> <ol style="list-style-type: none"> <li>Log in to the BCS console, click <b>Instance Management</b> in the navigation pane, and click an instance to go to the instance details page.</li> <li>Click the value next to <b>Cluster</b> to go to the CCE console, and click the target cluster. On the <b>Storage</b> page, check whether the PVC bound to the peer exists and is normal. <ul style="list-style-type: none"> <li>If it does not exist or is abnormal, create a PVC and bind it to the BCS instance.</li> <li>If it exists, perform the following steps.</li> </ul> </li> <li>On the BCS instance details page, click the <b>Monitoring</b> tab and then the <b>Active</b> tab. Record the value of <b>name</b> in the <b>Resource Name</b> column.</li> </ol> <p><b>Figure 2-11</b> Checking name of the peer that failed to access the database</p>  <pre> clusterName=... clusterID=cfb2...00029 kind=Pod namespace=default name=ord...2ca-0 uid=9f...b320b80 </pre> <ol style="list-style-type: none"> <li>Click the alarm and record <b>clusterID</b> and <b>name</b>.</li> <li>Go to the CCE console, click <b>Storage</b>, and check whether the PVC bound to the peer exists. If it does not exist, create a PVC and bind it to the peer.</li> <li>Log in to all nodes (bound with EIPs) in the CCE cluster where the instance is deployed and run the <b>docker ps   grep name</b> command (as shown in the following figure). The container whose name starts with <b>k8s_peer</b> (or <b>k8s_orderer</b> for an orderer) is the container for which the alarm is generated. The container ID is at the start of the section.</li> </ol>

Alarm Name	Alarm Source	Solution
		<p><b>Figure 2-12 Viewing the command output</b></p>  <p><b>NOTE</b> For details about how to log in to a node in a CCE cluster, see <a href="#">Viewing O&amp;M Logs on a Backend VM</a>.</p> <ol style="list-style-type: none"> <li>Run the <b>docker exec -it container id /bin/bash</b> command to enter the container.</li> <li>Run the <b>find / -name production</b> command to go to the found path, as shown in the following figure.</li> </ol> <p><b>Figure 2-13 Viewing the path</b></p>  <p>Check whether the <b>CURRENT</b>, <b>LOG</b>, and <b>MANIFEST-000***</b> files exist in the <b>ledgersData/stateLeveldb/</b> directory. If these files do not exist, run the <b>docker restart Container ID</b> command to restart the peer container.</p> <ol style="list-style-type: none"> <li>If the fault persists, go to <b>Log &gt; Log Files</b> on the AOM console. Download the log files of the peer and orderer on the cluster for which the alarm is generated, and send the log files to technical support.</li> </ol>

Alarm Name	Alarm Source	Solution
PeerNodeDiskAvailableNotEnough	BCS	<p>The peer disk space is insufficient and needs to be expanded. Perform the following steps to expand the disk space:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console, click <b>Instance Management</b> in the navigation pane, and click an instance to go to the instance details page.</li> <li>2. Click the <b>Monitoring</b> tab and then the <b>Active</b> tab. Record the value of <b>uid</b> in the <b>Resource Name</b> column.</li> </ol> <p><b>Figure 2-14</b> Checking uid</p>  <pre> clusterName=... clusterID=...029 kind=Pod namespace=default name=...bf52ca-0 uid=9f5t...6b320b80     </pre> <ol style="list-style-type: none"> <li>3. Log in to all nodes (bound with EIPs) in the CCE cluster where the BCS instance is deployed and run the <b>docker ps</b> command on the nodes one by one until you find the <b>Container ID</b>, that is, the first 12 digits of the uid obtained in the previous step. Record the value of the corresponding <b>NAMES</b>.</li> </ol> <p><b>Figure 2-15</b> Viewing the command output</p>  <pre> root@node01:~# docker ps CONTAINER ID   IMAGE     COMMAND                  STATUS    PORTS k8s_peer_...  hyperledger/fabric-peer  /peer start -- ...  Up 12 minutes k8s_peer_...  hyperledger/fabric-peer  /peer start -- ...  Up 5 hours k8s_peer_...  hyperledger/fabric-peer  /peer start -- ...  Up 5 hours k8s_peer_...  hyperledger/fabric-peer  /peer start -- ...  Up 5 hours k8s_peer_...  hyperledger/fabric-peer  /peer start -- ...  Up 5 weeks k8s_peer_...  hyperledger/fabric-peer  /peer start -- ...  Up 5 weeks     </pre> <p>For example, if the value of <b>NAMES</b> is <b>k8s_peer_peer-b738403d592c78f5463a8ccf24b4f7f8cc83d07a-0_default_b28328a1-8d7d-4686-8b4f-dd4333e2b400_0</b>, the corresponding peer name is <b>peer_peer-b738403d592c78f5463a8ccf24b4f7f8cc83d07a-0</b>.</p> <p><b>NOTE</b> For details about how to log in to a node in a CCE cluster, see <a href="#">Viewing O&amp;M Logs on a Backend VM</a>.</p> <ol style="list-style-type: none"> <li>4. On the BCS instance details page, click <b>More</b> on the <b>Basic Information</b> tab page and then click <b>View Details</b> next to <b>Network Storage</b> to obtain <b>PVC Name</b>.</li> <li>5. Log in to the CCE console, and choose <b>Resource Management</b> &gt; <b>Storage</b> in the navigation pane.</li> </ol>

Alarm Name	Alarm Source	Solution
		6. On the <b>SFS Turbo</b> tab page, select the target BCS instance's cluster, and click <b>Expand Capacity</b> in the row containing the recorded PVC.





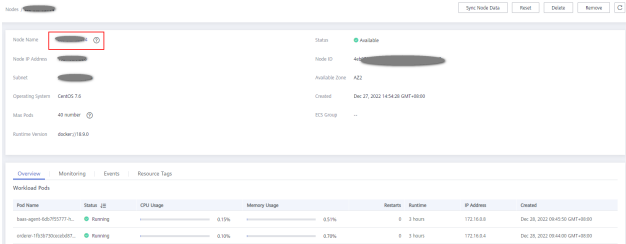
Alarm Name	Alarm Source	Solution
OrdererNode DiskAvailableNotEnough	BCS	<p>The orderer disk space is insufficient and needs to be expanded. Perform the following steps to expand the disk space:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console, click <b>Instance Management</b> in the navigation pane, and click an instance to go to the instance details page.</li> <li>2. Click the <b>Monitoring</b> tab and then the <b>Active</b> tab. Record the value of <b>uid</b> in the <b>Resource Name</b> column.</li> </ol> <p><b>Figure 2-16</b> Checking uid of the orderer</p>  <p><b>Figure 2-17</b> Checking the value of NAMES</p>  <p>For example, if the value of <b>NAMES</b> is <b>k8s_orderer_orderer-759ca9423d9805ed7b9b4aa274e54a2481aaaa5-0_default_2827a84a-dfbd-49b0-a96e-9a73b0c65a32_0</b>, the corresponding orderer name is <b>orderer_orderer-759ca9423d9805ed7b9b4aa274e54a2481aaaa5-0</b>.</p> <p><b>NOTE</b> For details about how to log in to a node in a CCE cluster, see <a href="#">Viewing O&amp;M Logs on a Backend VM</a>.</p> <ol style="list-style-type: none"> <li>4. On the BCS instance details page, click <b>More</b> on the <b>Basic Information</b> tab page and then click <b>View Details</b> next to <b>Network Storage</b> to obtain <b>PVC Name</b>.</li> <li>5. Log in to the CCE console, and choose <b>Resource Management</b> &gt; <b>Storage</b> in the navigation pane.</li> </ol>

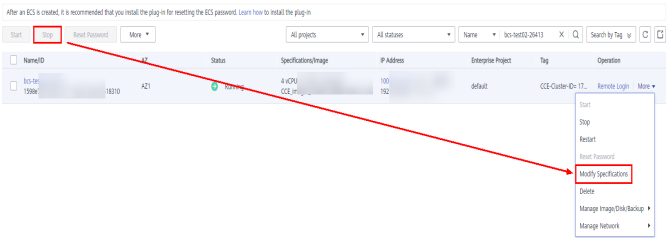
Alarm Name	Alarm Source	Solution
		6. On the <b>SFS Turbo</b> tab page, select the target BCS instance's cluster, and click <b>Expand Capacity</b> in the row containing the recorded PVC.
FailedPullImage	CCE	<p>The image address is incorrect. For example, the image address configured in the add-on at some sites is incorrect, or the permission configured for the image repository is incorrect.</p> <p>If a large number of images are pulled concurrently, some images may fail to be pulled. If the images can be pulled successfully after retry, the alarm is cleared.</p>
BackOffPullImage	CCE	<p>The image address is incorrect. For example, the image address configured in the add-on at some sites is incorrect, or the permission configured for the image repository is incorrect. If the images can be pulled successfully after retry, the alarm is cleared.</p>
FailedCreate	CCE	<p>Check the pod status of baas-agent, peer, and orderer. Do as follows:</p> <ol style="list-style-type: none"> <li>1. Check whether the pod scheduling policy is correct. Log in to the CCE console, choose <b>Workloads &gt; Deployments</b> or <b>StatefulSets</b> in the navigation pane, click the workload name to go to the workload details page, and check CPU requests and memory requests on the <b>Pods</b> tab.</li> <li>2. Check whether the node resources are sufficient. In the navigation pane, choose <b>Resource Management &gt; Nodes</b>. In the <b>Allocatable</b> column, view the available CPUs and memory size of the node where the pod is located.</li> </ol>
BackOffStart	CCE	<p>Check the pod status of baas-agent, peer, and orderer. Do as follows:</p> <ol style="list-style-type: none"> <li>1. Check whether the pod scheduling policy is correct. Log in to the CCE console, choose <b>Workloads &gt; Deployments</b> or <b>StatefulSets</b> in the navigation pane, click the workload name to go to the workload details page, and check CPU requests and memory requests on the <b>Pods</b> tab.</li> <li>2. Check whether the node resources are sufficient. In the navigation pane, choose <b>Resource Management &gt; Nodes</b>. In the <b>Allocatable</b> column, view the available CPUs and memory size of the node where the pod is located.</li> </ol>

Alarm Name	Alarm Source	Solution
Unhealthy	CCE	<p>Check the pod status of baas-agent, peer, and orderer. Do as follows:</p> <p>Log in to the CCE console, choose <b>Workloads &gt; Deployments</b> or <b>StatefulSets</b> in the navigation pane, click the workload name to go to the workload details page, and click <b>Upgrade &gt; Advanced Settings &gt; Health Check</b>.</p>
FailedScheduling	CCE	<p>Check the pod status of baas-agent, peer, and orderer. Do as follows:</p> <ol style="list-style-type: none"> <li>1. Check whether the node resources are sufficient. In the navigation pane, choose <b>Resource Management &gt; Nodes</b>. In the <b>Allocatable</b> column, view the available CPUs and memory size of the node where the pod is located.</li> <li>2. Check whether the pod scheduling policy is correct. Log in to the CCE console, choose <b>Workloads &gt; Deployments</b> or <b>StatefulSets</b> in the navigation pane, click the workload name to go to the workload details page, and check custom scheduling policies on the <b>Scheduling Policies</b> tab.</li> </ol> <p><b>NOTE</b> The coredns add-on is a DNS server that provides domain name resolution services for Kubernetes clusters. coredns chains plug-ins to provide additional features. At least two nodes are required to ensure the proper running of coredns. Therefore, if the number of nodes in the cluster where the BCS instance is located is less than 2, the alarm indicating failed scheduling is frequently generated. This alarm does not affect BCS functions.</p> <p>Do as follows:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console.</li> <li>2. In the navigation pane, click <b>Instance Management</b>.</li> <li>3. Click an instance name to go to the instance details page.</li> <li>4. On the <b>Monitoring</b> tab page, locate the row that contains the alarm, hover the mouse pointer over the resource name, and check the value of <b>name</b>. If the value starts with "coredns-", the alarm does not need to be handled.</li> </ol>

Alarm Name	Alarm Source	Solution
Rebooted	CEE	<p>The node has been restarted. If the baas-agent, peer, and orderer services are deployed on the node, check whether the pod status is abnormal. If these instances are not deployed on the node, BCS is not affected.</p> <p>Do as follows:</p> <ol style="list-style-type: none"> <li>1. Check whether the restart is caused by manual operations (such as shutdown and restart).</li> <li>2. Check whether the restart is caused by node resource overload. Go to the AOM console, choose <b>Monitoring &gt; Host Monitoring</b> in the navigation pane, and check the CPU usage and memory usage.</li> </ol>
NodeNotReady	CEE	<p>If the baas-agent, peer, and orderer services are deployed on the node, restore the node status or migrate services to other nodes.</p> <p>Do as follows:</p> <ol style="list-style-type: none"> <li>1. Check whether the node resources are sufficient. In the navigation pane, choose <b>Resource Management &gt; Nodes</b>. In the <b>Allocatable</b> column, view the available CPUs and memory size of the node where the pod is located.</li> <li>2. Restart the node.</li> <li>3. In the navigation pane, choose <b>Resource Management &gt; Nodes</b>. In the <b>Operation</b> column, choose <b>More &gt; Reset</b>.</li> </ol>

Alarm Name	Alarm Source	Solution
High Memory Usage on the Node	BCS	<p>If the memory usage exceeds 80%, the possible causes are as follows:</p> <ol style="list-style-type: none"> <li>1. There are too many transaction requests in a short time.</li> <li>2. The memory capacity of the node where the container is located cannot meet what is required by the instance specifications.</li> </ol> <p>Do as follows:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console. In the navigation pane, click <b>Instance Management</b>.</li> <li>2. Click an instance name to go to the instance details page.</li> <li>3. On the BCS instance details page, click the <b>Monitoring</b> tab and then the <b>Active</b> tab. Record the value of <b>name</b> in the <b>Resource Name</b> column.</li> </ol> <p><b>Figure 2-18</b> Checking the value of name of the peer</p>  <p>The screenshot shows a terminal window with the following text:         <pre>         clusterName=...         clusterID=cfb2...00029         kind=Pod         namespace=default         name=...52ca-0         uid=9f...320b80         </pre>         The 'name=' line is highlighted with a red box.       </p> <ol style="list-style-type: none"> <li>4. Go to the CCE console and locate the cluster where the abnormal node is. On the <b>Resource Management &gt; Nodes</b> page, click the node name to go to the node details page. Click the node name to go to the ECS page.</li> <li>5. Stop the ECS, and then choose <b>More &gt; Modify Specifications</b>. Select a new flavor with desired memory.</li> </ol>

Alarm Name	Alarm Source	Solution																					
Excessive memory usage	BCS	<p>If the memory usage exceeds 90%, the possible causes are as follows:</p> <ol style="list-style-type: none"> <li>1. There are too many transaction requests in a short time.</li> <li>2. The memory capacity of the node where the container is located cannot meet what is required by the instance specifications.</li> </ol> <p>Do as follows:</p> <ol style="list-style-type: none"> <li>1. Log in to the BCS console. In the navigation pane, click <b>Instance Management</b>.</li> <li>2. Click an instance name to go to the instance details page.</li> <li>3. On the BCS instance details page, click the <b>Monitoring</b> tab and then the <b>Active</b> tab. Record the value of <b>name</b> in the <b>Resource Name</b> column.</li> </ol> <p><b>Figure 2-19</b> Checking the value of name</p>  <pre> clusterName=... clusterID=cfb2...00029 kind=Pod namespace=default name=prd...52ca-0 uid=9f...320b80     </pre> <ol style="list-style-type: none"> <li>4. Go to the CCE console and locate the cluster where the abnormal node is. On the <b>Resource Management &gt; Nodes</b> page, click the node name to go to the node details page. Click the node name to go to the ECS page.</li> </ol> <p><b>Figure 2-20</b> Node details page</p>  <p>The screenshot shows the Node details page with the following information:</p> <ul style="list-style-type: none"> <li>Node Name: [Redacted]</li> <li>Node ID: [Redacted]</li> <li>Node IP Address: [Redacted]</li> <li>Subnet: [Redacted]</li> <li>Operating System: CentOS 7.6</li> <li>Max Pods: 41 number</li> <li>Instance Version: [Redacted]</li> <li>Status: Available</li> <li>Node ID: [Redacted]</li> <li>Available Zone: AZ2</li> <li>Created: Dec 27, 2022 14:54:26 GMT+08:00</li> <li>ECS Group: [Redacted]</li> </ul> <p>Below the details is a table for Workload Pods:</p> <table border="1"> <thead> <tr> <th>Pod Name</th> <th>Status</th> <th>CPU Usage</th> <th>Memory Usage</th> <th>Restart</th> <th>IP Address</th> <th>Created</th> </tr> </thead> <tbody> <tr> <td>name-9p9t-6673077171</td> <td>Running</td> <td>0.0%</td> <td>0.0%</td> <td>0</td> <td>192.168.1.1</td> <td>Dec 28, 2022 09:47:51 GMT+08:00</td> </tr> <tr> <td>name-1f03733000807</td> <td>Running</td> <td>0.0%</td> <td>0.0%</td> <td>0</td> <td>192.168.1.2</td> <td>Dec 28, 2022 09:48:02 GMT+08:00</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>5. Stop the ECS, and then choose <b>More &gt; Modify Specifications</b>. Select a new flavor with desired memory.</li> </ol>	Pod Name	Status	CPU Usage	Memory Usage	Restart	IP Address	Created	name-9p9t-6673077171	Running	0.0%	0.0%	0	192.168.1.1	Dec 28, 2022 09:47:51 GMT+08:00	name-1f03733000807	Running	0.0%	0.0%	0	192.168.1.2	Dec 28, 2022 09:48:02 GMT+08:00
Pod Name	Status	CPU Usage	Memory Usage	Restart	IP Address	Created																	
name-9p9t-6673077171	Running	0.0%	0.0%	0	192.168.1.1	Dec 28, 2022 09:47:51 GMT+08:00																	
name-1f03733000807	Running	0.0%	0.0%	0	192.168.1.2	Dec 28, 2022 09:48:02 GMT+08:00																	

Alarm Name	Alarm Source	Solution
		<p><b>Figure 2-21</b> Modifying specifications</p> 

## Viewing Alarms

- Step 1** Log in to the BCS console.
- Step 2** In the navigation pane, click **Instance Management** to view the basic information of a BCS instance, including the blockchain type, consensus mechanism, status, and creation time.
- Step 3** On an instance card, click the instance name.
- Step 4** Click the **Monitoring** tab to view alarms generated in BCS and CCE. In the upper right corner, you can filter alarms generated in the last 30 minutes, 1 hour, or 1 day, or search for a specified alarm.
- Step 5** Click an alarm to view its details. Alarm sources include BCS and CCE. For details about how to handle alarms, see [Table 2-9](#).

----End

### 2.4.2.3 Setting Web Disk Space Alarms

#### Introduction

BCS is connected to AOM. AOM is a one-stop platform for technical support to monitor the application and resource operating state in real time. By analyzing metrics, alarms, and logs, you can quickly locate root causes to ensure smooth running of services.

The following describes how to use AOM to monitor the disk status of a BCS instance.

#### Setting Alarms

When technical support needs to check the web disk metrics, they can use the AOM service to set alarm rules for the disk metrics. If a metric exceeds the threshold, the system automatically sends an alarming SMS message or email.

**Step 1** Log in to the SMN console, create a topic and add subscription.

If you need to obtain resource change information in real time, create a topic and add subscribers to this topic. In this way, the email addresses or mobile numbers of recipients are noted by the system. When establishing rules, you can select the relevant recipient.

1. Create a topic.
2. Add subscription to the topic.

**Step 2** Go to the AOM console to create alarm rules.

1. In the navigation pane, choose **Alarm Center** > **Alarm Rules**. Then, click **Create Alarm Rule**.
2. Set basic information such as the rule name and description.
3. Set **Rule Type** to **Threshold alarm**, set **Monitored Object** and **Alarm Condition**, and click **Create Now**. For details, see [Creating Alarm Rules](#).

----End

#### 2.4.2.4 Disk Metrics

After metric thresholds and alarming criteria related to disk usage are configured, alarming short messages or emails can be sent to technical support. In this way, technical support can detect and handle service exceptions in a timely manner to reduce the loss caused by exceptions. The following table lists the metrics related to disks used for BCS services.

**Table 2-10** Node metrics

Metrics	Description	Meaning	Value Range	Unit
diskAvailableCapacity	Available disk space	Disk space that is not used	$\geq 0$	MB
diskCapacity	Disk capacity	Total disk capacity	$\geq 0$	MB
diskReadRate	Disk read rate	Data volume read from the disk per second	$\geq 0$	KB/s
diskRWStatus	Disk read/write status	Read/write status of the disk on a node	<b>0</b> (read and write) and <b>1</b> (read-only).	None
diskUsedRate	Disk usage	Percentage of the used disk space to the total disk space	$\geq 0$	Percentage



Metrics	Description	Meaning	Value Range	Unit
diskWriteRate	Disk write rate	Data volume written into the disk per second	$\geq 0$	KB/s

Disk metrics can be calculated on the following basis.

**Table 2-11** Metric measurement bases

Basis	Description
clusterId	Cluster ID
clusterName	Cluster name
hostID	Node ID
namespace	Cluster namespace
nodeIP	IP addresses of a node
nodeName	Node name

## 2.4.2.5 Viewing O&M Logs

### Introduction

If an exception occurs when you use a BCS instance, view the O&M logs to analyze and locate the fault for quick rectification. This section describes how to view the O&M logs of each BCS instance node in the CCE cluster on the frontend GUI and backend virtual machines (VMs).

**Table 2-12** BCS instance logs

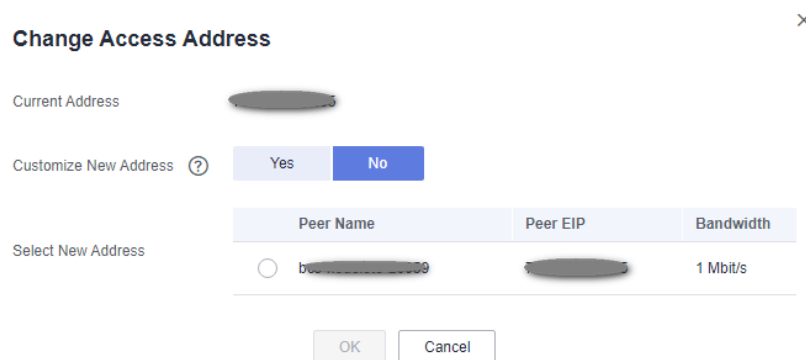
Component	Description	Log Path
baas-agent	Blockchain management run log	<code>/var/paas/sys/log/baas-agent/baas-agent.log</code> <code>/var/paas/sys/log/baas-agent/audit.log</code>
peer	Peer run log	<code>/var/paas/sys/log/baas-service/peer/audit.peer-*****-*.log</code> <code>/var/paas/sys/log/baas-service/peer/peer-*****-*.trace</code>

Component	Description	Log Path
orderer	Orderer run log	/var/paas/sys/log/baas-service/orderer/audit.orderer-*****-.log /var/paas/sys/log/baas-service/orderer/orderer-*****-*-start.trace /var/paas/sys/log/baas-service/orderer/orderer-*****-*.trace

## Viewing O&M Logs on a Backend VM

- Step 1** On the CCE console, view and record the node name on the **Workloads** page.
- Step 2** On the **Instance Management** page of the BCS console, locate the instance and choose **More > Change Access Address** to view the access address.

**Figure 2-22** Viewing the access address

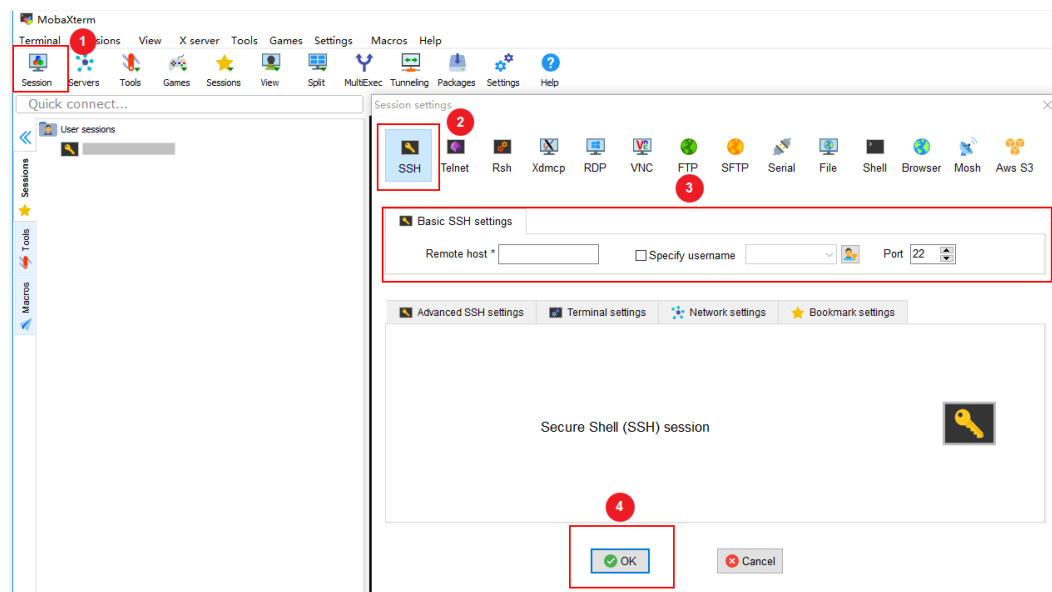


**NOTE**

The node where the BCS instance is deployed must be bound with an EIP.

- Step 3** Log in to the VM corresponding to the access address, and view the O&M logs.

Figure 2-23 Logging to the VM



Enter the VM address (the access address obtained in [Step 2](#)) for **Remote host**, and enter the VM username for **Specify username**.

1. Check baas-agent node logs.
  - a. Run the following command to query the baas-agent node ID:  
docker ps|grep baas-agent

Figure 2-24 Checking the baas-agent node ID

```
[root@log-1t-44243 ~]# docker ps |grep baas-agent
0b2911c07a7b    db11e1933c3d    "/bin/bash -c 'exp..." 2 days ago    Up 2 day
s
1efddfa0d7bd    cfe-pause:11.23.1    k8s_POD_baas-agent-9885db668-mxvm4_default_933fee36-b356-11e9-b003-fa163ec54113_0    "/pause"    2 days ago    Up 2 day
s
k8s_POD_baas-agent-9885db668-mxvm4_default_933fee36-b356-11e9-b003-fa163ec54113_0
```

- b. Run the following command to query the baas-agent node logs:  
docker logs ID -f

Figure 2-25 Checking the baas-agent node logs

```
[root@log-1t-44243 ~]# docker ps |grep baas-agent
0b2911c07a7b    db11e1933c3d    "/bin/bash -c 'exp..." 2 days ago    Up 2 day
s
1efddfa0d7bd    cfe-pause:11.23.1    k8s_POD_baas-agent-9885db668-mxvm4_default_933fee36-b356-11e9-b003-fa163ec54113_0    "/pause"    2 days ago    Up 2 day
s
k8s_POD_baas-agent-9885db668-mxvm4_default_933fee36-b356-11e9-b003-fa163ec54113_0
[root@log-1t-44243 ~]# docker logs -f 0b2911c07a7b
The make env.sh user is root, bcsid is 18636745-821a-cf15-5abd-152ee3b7115b
chown: changing ownership of '/opt/gopath/src/github.com/hyperledger/fabric/orderer/crypto/ordererOrganizations/orderer-89c01f73e87fd64b084f2716aa050925c20860cb-admin/admin/.data': Read-only file system
chown: changing ownership of '/opt/gopath/src/github.com/hyperledger/fabric/orderer/crypto/ordererOrganizations/orderer-89c01f73e87fd64b084f2716aa050925c20860cb-admin/admin/tls': Read-only file system
chown: changing ownership of '/opt/gopath/src/github.com/hyperledger/fabric/orderer/crypto/ordererOrganizations/orderer-89c01f73e87fd64b084f2716aa050925c20860cb-admin/admin/msp': Read-only file system
chown: changing ownership of '/opt/gopath/src/github.com/hyperledger/fabric/orderer/crypto/ordererOrganizations/orderer-89c01f73e87fd64b084f2716aa050925c20860cb-admin/admin/...' 2019-07-31 05:46:43.468296142711s/server-key': Read-only file system
```

2. Check the logs of a peer node.
  - a. Run the following command to query the peer node ID:  
docker ps|grep peer

Figure 2-26 Checking the peer ID

```
[root@master1-5801 ~]# docker ps |grep peer
c8c79838887a    ka1a0c-f1411    "/bin/bash -c 'expor..." 3 weeks ago    Up 3 weeks    k8s_peer_58aea736051a6156347ae3bc12212980298_1_default_850a1c5e-a84a-4161-8bc2-c307394
892f2f0    ka1a0c-f1411    "/bin/bash -c 'expor..." 3 weeks ago    Up 3 weeks    k8s_peer_9546e90844662a3869447b7022af5334ab5ee_1_default_b3a1201f-9b08-4c43-a46a-7145cd2
c8e1c9b73a    ka1a0c-f1411    "/bin/bash -c 'expor..." 3 weeks ago    Up 3 weeks    k8s_peer_9546e90844662a3869447b7022af5334ab5ee_1_default_b3a1201f-9b08-4c43-a46a-7145cd2
8095f9    cce-pause:3.1    "/pause"    3 weeks ago    Up 3 weeks    k8s_POD_peer_58aea736051a6156347ae3bc12212980298_1_default_850a1c5e-a84a-4161-8bc2-c307394
```

- b. Run the following command to query the peer node logs:  
docker logs -f ID

Figure 2-27 Checking the peer logs

```
[root@master01-54891 ~]# docker logs -f 58c7683b87a
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/core.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/.2021_02_02_09_34_59_413905497/core.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/.2021_02_02_09_34_59_413905497': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/impmapping.json': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/.2021_02_02_09_34_59_565881316/impmapping.json': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/.2021_02_02_09_34_59_565881316': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/core.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/.2021_02_02_09_34_59_413905497/core.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/.2021_02_02_09_34_59_413905497': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_fabriccoreconfigmap/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/impmapping.json': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/.2021_02_02_09_34_59_565881316/impmapping.json': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping/.2021_02_02_09_34_59_565881316': Read-only file system
chown: changing ownership of '/etc/hyperledger/impmapping': Read-only file system
++ hostname
+ H05NWE-peer-50aea7369551a6c15634c7ae3bc12212980298-1
+ sed -i '/fileSystemPath: V:/var/hyperledger/production/c\ fileSystemPath: /home/paas/evs/baas/1a518637-0a63-6e67-253b-5846420c45fc/peer-50aea7369551a6c15634c7ae3bc12212980298-1/production' core.yaml
+ sed -i '/id: jdoe/c\ id: peer-50aea7369551a6c15634c7ae3bc12212980298-1' core.yaml
+ sed -i '/localMspId: BFA437/c\ localMspId: 50aea7369551a6c15634c7ae3bc12212980298P' core.yaml
++ 'l' -z 32623 'l'
++ sed -i '/address: 0.0.0.0:7051/c\ address: peer-50aea7369551a6c15634c7ae3bc12212980298-1.peer-50aea7369551a6c15634c7ae3bc12212980298.default.svc.cluster.local:32624' core.yaml
++ /bin/ip route get 1.2.3.4
++ head -1
++ tr -s ' '
++ cut -d ' ' -f7
+ localIP=0.0.207
```

3. Check the logs of an orderer node.

- a. Run the following command to query the orderer ID:  
docker ps|grep orderer

Figure 2-28 Checking the orderer ID

```
[root@mf-test-60988 ~]# docker ps|grep orderer
77daf8baf444      89f4ba19145e      /bin/bash -c 'H05...'   2 days ago        Up 2 days
orderer_orderer-6f8ddd01fb38c8dff68ecdc9bb5d97e27df1ecf-0 default_B167d0d7-b750-11e9-bdf7-fa163e730475_0
```

- b. Run the following command to query the orderer logs:  
docker logs -f ID

Figure 2-29 Checking the orderer logs

```
[root@mf-test-60988 ~]# docker logs 77daf8baf444
chown: changing ownership of '/etc/hyperledger/configtx/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/genesis.block': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/.2019_08_05_07_13_20_605881597/genesis.block': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/.2019_08_05_07_13_20_605881597': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/orderer.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/.2019_08_05_07_13_20_296162454/orderer.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/.2019_08_05_07_13_20_296162454': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/genesis.block': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/.2019_08_05_07_13_20_605881597/genesis.block': Read-only file system
chown: changing ownership of '/etc/hyperledger/configtx/.2019_08_05_07_13_20_605881597': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/.data': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/orderer.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/.2019_08_05_07_13_20_296162454/orderer.yaml': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap/.2019_08_05_07_13_20_296162454': Read-only file system
chown: changing ownership of '/etc/hyperledger/temp_ordererconfigmap': Read-only file system
```

----End

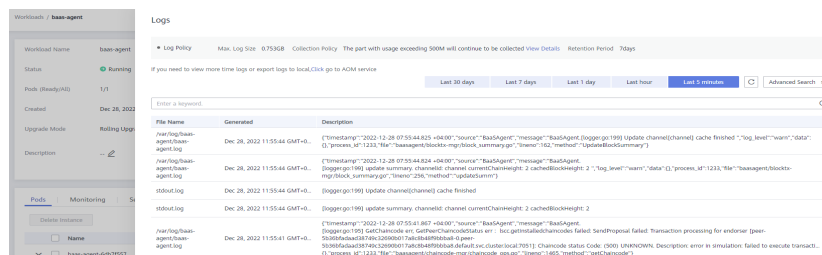
### 2.4.2.6 Viewing Chaincode Debug Logs

You can view chaincode debug logs to analyze and locate problems. This section describes how to view chaincode debug logs on the CCE console.

#### Procedure

- Step 1** Log in to the CCE console.
- Step 2** In the navigation pane, choose **Workloads > Deployments**. Select the cluster where the BCS instance is deployed.
- Step 3** Click the workload whose name starts with **baas-agent**.
- Step 4** Click **Logs** in the upper right corner to view the logs of the chaincode container. To view more logs or export logs, go to the AOM console.

Figure 2-30 Viewing the chaincode pod logs



----End

## 2.5 Channel Management

Peers communicate through channels. You can create channels and add organizations and peers to them.

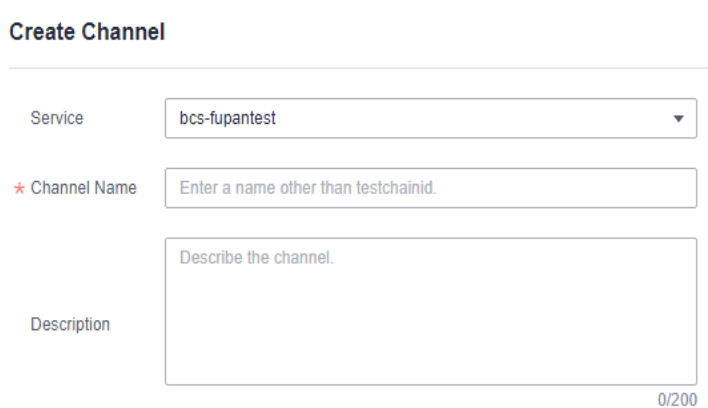
### Creating a Channel

- Step 1** Log in to the BCS console.
- Step 2** Click **Channel Management** in the navigation pane on the left. Click **Create Channel** in the upper right corner of the page.

**NOTE**

- The maximum number of channels for each instance is 2 for the basic edition and 4 for the professional edition.
- In a consortium, channels cannot be created for invitees' instances.

- Step 3** Select an instance, enter a channel name and description, and click **OK**.



----End

### Managing Channel Organizations and Peers

**NOTE**

This operation is not supported for invitees.

- Step 1** After the channel is created, click **Manage Organization and Peer** in the **Operation** column of the channel list.

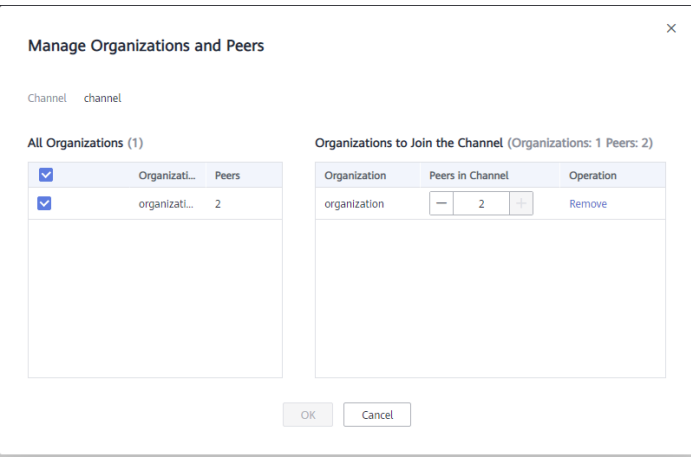
**Step 2** Select organizations and specify the number of peers you want to add to the channel.

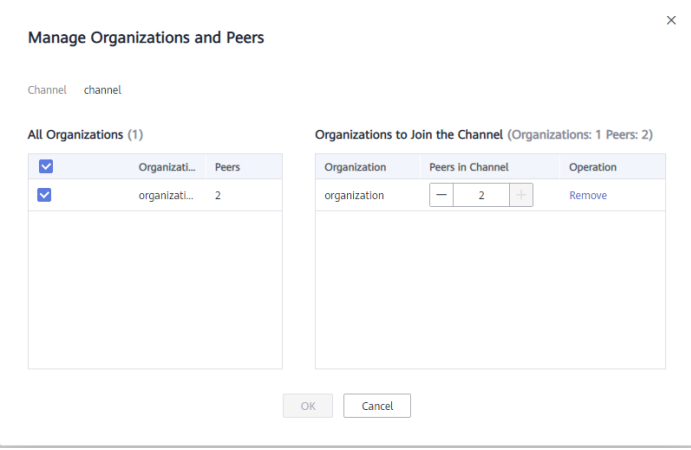
**Step 3** Click **OK**.

----End

## Other Operations

**Table 2-13** Other operations

Operation	Description
Searching for a channel	Enter a channel name in the search box at the upper right corner of the <b>Channel Management</b> page to search for the channel.
Querying channels	A channel list is displayed on the <b>Channel Management</b> page. You can view the channel name, instance name, and the channel nodes.
Viewing a peer	Click <b>View Peer</b> in the <b>Operation</b> column of the channel list to view peer information by organization, including the Membership Service Provider (MSP) ID, peer details (name, IP address, port, and domain), and whether the peer has been added to the channel.
Removing peers in an organization from a channel	<p>Click <b>Manage Organization and Peer</b> in the <b>Operation</b> column of the channel list. Decrease the value for <b>Peers in Channel</b> under <b>Organizations to Join the Channel</b>, then click <b>OK</b> to remove peers from the channel.</p> <p><b>Figure 2-31</b> Managing organizations and peers</p>  <p><b>NOTE</b> Keep at least <b>1</b> peer in the channel. To remove an organization from the channel, you can manually set the number of peers in the channel to <b>0</b>.</p>

Operation	Description
<p>Removing organizations from a channel</p>	<p>Click <b>Manage Organization and Peer</b> in the <b>Operation</b> column of the channel list. Under <b>Organizations to Join the Channel</b>, click <b>Remove</b> in the row that contains the target organization, then click <b>OK</b> to remove the organization from the channel.</p> <p><b>Figure 2-32</b> Removing organizations from a channel</p>  <p><b>NOTE</b> If an organization is listed in the endorsement policy of a chaincode, update the endorsement policy after the organization is removed from the channel. Otherwise, transactions will fail. For details, see <a href="#">Chaincode Management</a>.</p>
<p>Deleting a channel</p>	<p>Click <b>Delete</b> in the <b>Operation</b> column, then click <b>OK</b>.</p> <p><b>NOTE</b> Clear all organization nodes in a channel before you delete it.</p>

## 2.6 Blockchain Management

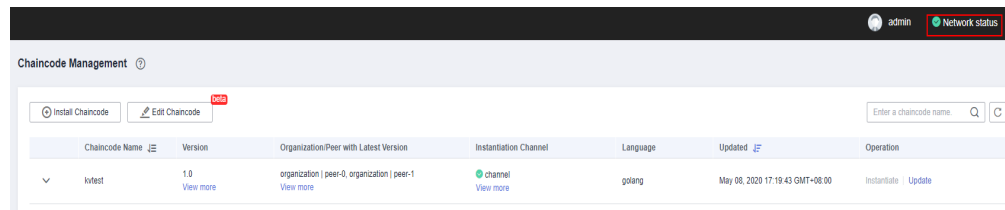
### 2.6.1 Chaincode Management

You can install, instantiate, and update chaincodes on the web. You can also check the Golang chaincode security during installation and update.

#### Note

1. Before installing a chaincode, compress the chaincode file into a .zip package.
2. If the **Network Status** displayed in the upper right corner of the **Blockchain Management** page is abnormal, do not perform any operations. Wait for a few minutes until the network is recovered.

Figure 2-33 Normal network status



## Installing a Chaincode

**Step 1** Log in to the **Blockchain Management** console. Click **Manage Blockchain** on an instance card. Enter the username and password, and click **Log In**.

**NOTE**

The username is **admin**, and the initial login password is the resource access initial password set when you deploy the BCS instance. To ensure system security, change the password periodically.

**Step 2** On the **Chaincode Management** page, click **Install Chaincode**.

**Step 3** Specify the chaincode name, version, and other parameters by referring to [Table 2-14](#).

Figure 2-34 Installing a chaincode

**Install Chaincode**

\* Chaincode Name

\* Chaincode Version

Ledger Storage

Select All Peers

Organization & Peer  ▼

Language  ▼

Chaincode File

Chaincode Description   
0/500

Code Security Check  ?



**Table 2-14** Chaincode parameters

Parameter	Description
Chaincode Name	Chaincode name, which can contain 6 to 25 including lowercase letters and digits, and must start with a letter.
Chaincode Version	Chaincode version.
Ledger Storage	Default option: <b>File database (goleveldb)</b> .
Select All Peers	Check the box to select all peers.
Organization & Peer	Manually select organizations and peers.
Language	Golang, Node.js, and Java are supported.
Chaincode File	Add a chaincode file.
Chaincode Description	Enter a description.
Code Security Check	This option is displayed only when the chaincode language is Golang. Enable this option to check code security.

**Step 4** Click **Install**.

**Step 5** Click  next to a chaincode name to view the details.

**Step 6** Click **Download** in the **Operation** column to view the check result. (The following example is for reference only.)

 **NOTE**

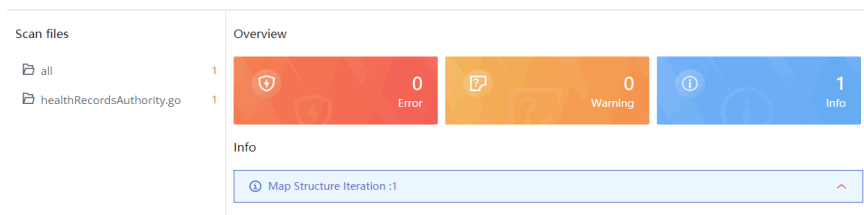
If **Code Security Check** is not enabled, no check report will be generated, and the **Download** button will not be displayed.

**Figure 2-35** Downloading the check report

Chaincode Version	SHA-256 Hash	Description	Installed	Operation
1.0	3455d016a31349749ff575c7b00c131782e130832a0a1afe61f...		Nov 15, 2021 10:26:23 GMT+08:00	<a href="#">Download</a> <a href="#">Delete</a>

- 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 2-36** Scanned files



2. For example, there is an info-level issue in the proceeding 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

After a chaincode is installed, it must be instantiated on the channel so that the peers can interact with each other using the distributed ledger and the chaincode container. Before instantiating a chaincode, add the peers to the channel. Otherwise, the chaincode cannot be instantiated.

**Step 1** Click **Instantiate** in the **Operation** column of the chaincode list.

**Step 2** Specify the channel for instantiation, chaincode version, endorsement policy, endorsing organizations, and chaincode parameters.

 **NOTE**

Endorsement is a process in which organizations perform a chaincode transaction and return a proposal response to a client application. An endorsement policy specifies how many members of different organizations on a channel are required to execute and validate a transaction based on the specified smart contract to make the transaction valid. Therefore, an endorsement policy defines the organization peers that must "endorse" (that is, approve of) the execution of a proposal.

- **Endorsement from any of the following organizations:** A transaction is valid as long as any one of the organizations endorses it.
- **Endorsement from all of the following organizations:** A transaction is valid only when all organizations endorse it.

**Figure 2-37** Instantiating a chaincode

**Instantiate Chaincode** [X]

Chaincode Name: kvtest001

Channel: channel

Chaincode Version: 2.0

\* Initialization Function: Enter a function, for example, init().  
Chaincode function that will be invoked

Chaincode Parameters: For example, a,200,b,250  
Enter the parameters of the initialization function init(). Separate multiple parameters with commas.

Endorsement Policy:  Endorsement from any of the following organizations  
 Endorsements from all of the following organizations

Endorsing Organizations: byl-ief-002, byl-ief-003

Privacy Protection Configuration:  No  Yes

Please input JSON data.  
For example:
 

```
{
  "name": "collectionPrivateDetails",
  "policy": "OR('Org1MSP.member', 'Org2MSP.member')",
  "requiredPeerCount": 0,
  "maxPeerCount": 3,
  "blockToLive": 0,
  "memberOnlyRead": true
}
```

 0/5,000

[Instantiate] [Cancel]

**Step 3** Enter the private data (JSON format) to be protected in the text box below **Privacy Protection**.

If you want to restrict data in a shared channel to certain specified members, use the privacy protection function. Skip this step if privacy protection is not required for your chaincode.

Configure privacy protection by referring to the example and the following parameter description:

- **name:** Name of the collection of private data, for example, collectionPrivateDetails.

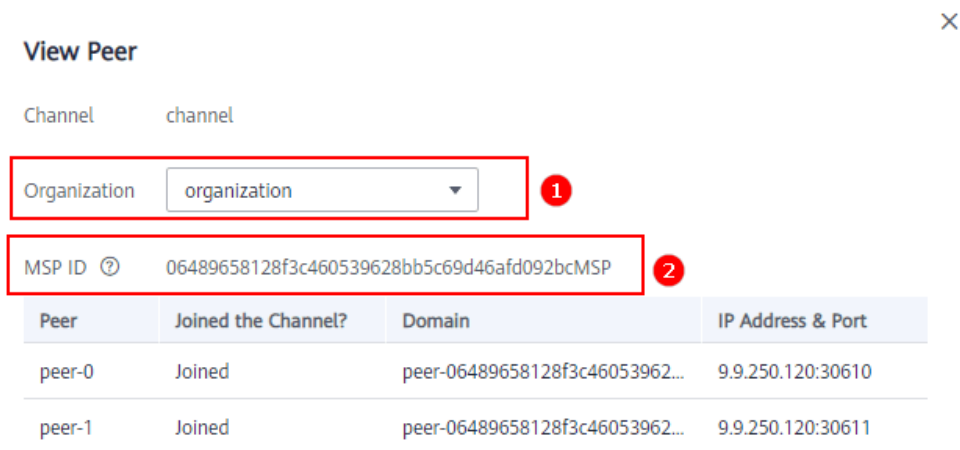
In a chaincode, if you want to write data to the collection of private data, ensure that the collection name is the same as that defined here.

```
stub.PutPrivateData("collectionPrivateDetails", key, value)
```

- **policy:** Peers allowed to access the data in the collection. In the example, only peers of organizations Org1 and Org2 are allowed to obtain the data in the collection.

Click **View Peer** on the **Channel Management** page, and obtain the MSP IDs of the two organizations, as shown in the following figure.

**Figure 2-38** Checking the MSP



- **requiredPeerCount:** Number of endorsing peers to which the private data can be disseminated. In the example, value **0** indicates that there is no endorsing peer.
- **maxPeerCount:** Maximum number of orderers, which is **3** in the example. Multiple orderers can be used for data redundancy. If one orderer is unavailable, other orderers can respond to requests for obtaining the private data.
- **blockToLive:** Maximum number of blocks that the private data can live for. If the number of blocks exceeds the threshold, the private data will be cleared. To keep private data indefinitely, set this parameter to **0**.
- **memberOnlyRead:** The default value is **true**. The access policy set in **policy** takes effect only when **memberOnlyRead** is set to **true**.

Example of privacy protection configuration (JSON):

```
[
  {
    "name": "collectionPrivateDetails",
    "policy": "OR('<Org1MSP>.member','<Org2MSP>.member')",
    "requiredPeerCount": 0,
    "maxPeerCount": 3,
    "blockToLive": 0,
    "memberOnlyRead": true
  }
]
```

This configuration indicates that the chaincode uses a private data space called **collectionPrivateDetails**. Only the peers of organizations Org1 and Org2 have access to the data in this space.

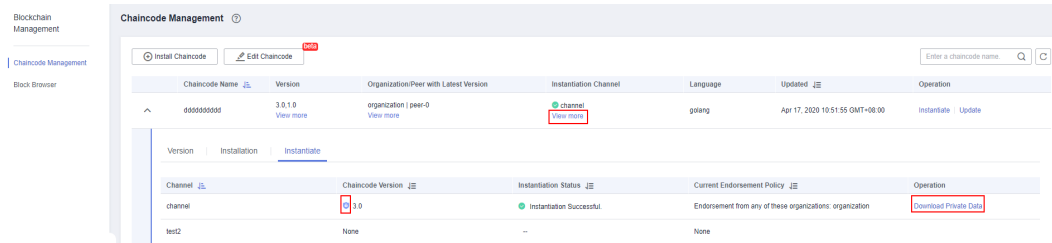
**NOTE**

The values of **name** and **blockToLive** cannot be modified during subsequent chaincode upgrade. For more information, see [Using Private Data in Fabric](#).

**Step 4** Click **Instantiate**.

If privacy protection is configured, you can click **View More** after the chaincode is successfully instantiated to download the private data and check whether the privacy protection settings are correct.

**Figure 2-39** Downloading private data




If chaincode instantiation fails, refer to [Chaincode Instantiation Error Codes](#) to determine the cause.

----End

## Updating a Chaincode

If your chaincode is updated, install and instantiate it again to meet new business requirements.

- Step 1** Click **Update** in the **Operation** column of the chaincode list.
- Step 2** Specify the chaincode version, select peers, add a chaincode file, and click **Update**.
- Step 3** Instantiate the updated chaincode. For details, see [Instantiating a Chaincode](#).
- Step 4** (Optional) Click  in front of the chaincode name. You can see details about this chaincode, including versions, and installation and instantiation information.

----End

## Chaincode Instantiation Error Codes

Chaincode instantiation may fail due to various causes. When confronted with an instantiation failure, you can refer to the following table to determine the cause.

**Table 2-15** Error codes

Error Code	Message
6001	Instantiation timed out.
6999	Unknown error.
6701	Client failed to connect to a peer.
6703	Endorsement signature failed verification.
6704	Failed to pull the ccenv image during chaincode compilation.
6705	Chaincode compilation failed.

Error Code	Message
6707	Failed to build a chaincode image.
6708	Failed to create a chaincode container.
6709	Failed to register the chaincode container.
6710	Client failed to connect to an orderer.
6712	Transaction recording in distributed ledgers failed.
6713	Request error determined by the orderer.
6714	The endorsement policy failed the verification.
6715	Instantiation failed because instantiation of another chaincode has already been started.
6716	Error detected in the init() function parameters.
6717	Error detected in the invoke() function parameters.
6720	Failed to create a chaincode certificate.
6721	Chaincode container startup timed out.
6722	Transaction timed out because init() execution abnormally terminates after startup of the chaincode container.
6723	A chaincode with the same schema has already been instantiated on this channel.
6725	The signature set does not satisfy the endorsement policy.
6726	The instantiation policy failed the verification. Select a peer of an organization that exists in the channel before chaincode instantiation to upgrade the chaincode.
6901	Instantiation failed. The chaincode to be instantiated must contain all the tables in the previously instantiated chaincode.
6902	Instantiation failed. The chaincode to be instantiated must contain all the fields in the previously instantiated chaincode.
6903	Instantiation failed. The chaincode to be instantiated must not contain any changes to the field attributes included in the previously instantiated chaincode.
6904	The schema file of the instantiated chaincode does not exist.
6905	Failed to resolve the schema file.
6906	Insufficient disk space.

## 2.6.2 Block Browser

You can query blockchain information required for maintenance, including the block quantity, transaction quantity, block details, transaction details, performance, and peer statuses.

 **NOTE**

To access blockchain browsers, set the blockchain network access address to a private address of the cluster and ensure that the network between the user and cluster is connected. If you set the access address to an EIP bound to the cluster, unbind the EIP when you are not using the blockchain browser.

### Procedure

**Step 1** Open the block browser page.

1. Log in to the BCS console.
2. Click **Manage Blockchain** on an instance card.
3. Enter the username and password and click **Log In**.
4. Click **Block Browser** in the navigation pane.

**Step 2** Select a channel from the **Channel** drop-down list box. Real-time data is displayed in the lower part of the page.

**Step 3** You can view the following data in the block browser.

**Table 2-16** Blockchain data

Item	Description
Peers	Number of peers in the selected channel
Chaincodes	Number of installed chaincodes
Blocks	Number of generated blocks
Transactions	Number of transactions that have been performed
Block details	Click the <b>Block List</b> tab to view the block hash and data hash of recent blocks.
Transaction list	<ul style="list-style-type: none"> <li>• Click the <b>Transaction List</b> tab to view the information about recent transactions such as the transaction IDs, creators' MSPs, and creation time.</li> <li>• Click <b>View Details</b> in the <b>Operation</b> column of the transaction list to view more details about the transaction.</li> </ul>

Item	Description
Performance analysis	<p>The line charts show the trends of performance data, helping you know the performance status.</p> <ul style="list-style-type: none"><li>• Block performance: Click <b>Block</b> to view changes in the block quantity. Move the pointer along the curve to view the number of blocks at different time points.</li><li>• Transaction performance: Click <b>Transaction</b> to view changes in the transaction quantity. Move the pointer along the curve to view the number of transactions at different time points.</li></ul> <p><b>NOTE</b> You can select a time granularity (hours or minutes) in the upper right corner of the chart.</p>
Transaction quantity of organizations	<p>The pie chart shows the percentage of each organization's transactions.</p> <p><b>NOTE</b> Move the pointer on the pie chart to view the transaction quantity and percentage of each organization.</p>
Peer statuses	<p>You can view the running statuses of all peers in the selected channel to detect exceptions of peers in time.</p>

----End

## 2.7 Downloading SDK Configurations and Certificates

BCS supports chaincode functions such as execution and query. Before developing an application, download the certificates and SDK configuration. The SDKs can use the configuration file to easily access the blockchain network and complete transactions. You do not need to manually configure the SDKs.

### Prerequisites

Before downloading the SDK configuration, ensure that the chaincode has been installed and instantiated.

### Downloading SDK Configurations and Certificates

The SDK configuration, certificates, and application must be used together. The SDK configuration file contains chaincode and certificate path information. Specify the chaincode name and the storage path of the downloaded certificate on the application executor when downloading the SDK configurations. If the certificate path changes, you must manually change all certificate paths in the SDK configuration file.

BCS supports three types of certificates: administrator certificate, user certificate, and CA certificate. The administrator certificate is required to create, join, and update a channel, and install, instantiate, update, and delete a chaincode. For transactions and query, you are advised to use the user certificate. Download the certificates on the **Instance Management** page.



- An administrator certificate contains the organization's administrator permission certificate and private key and can be used to manage channels and contracts.
- A user certificate contains the organization's user permission certificate and private key and can be used for transactions and queries.
- A CA certificate is the root certificate of an organization. The CA public and private key pair can be used to issue lower-level certificates.

 **NOTE**

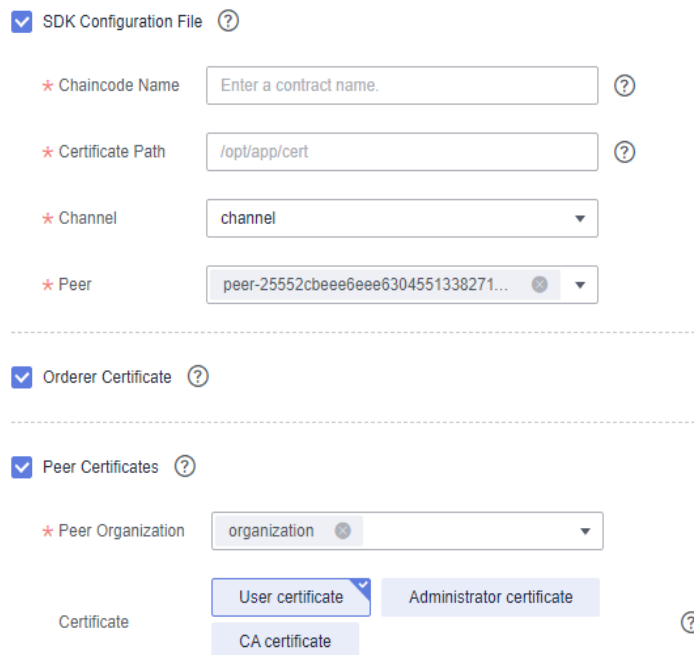
- The administrator certificate differs between an orderer and a peer. For management within a channel, use the administrator certificate for peers instead of that for orderers.
- Encrypt the private keys in the downloaded certificates for storage.


**Step 1** Log in to the BCS console.


**Step 2** In the navigation pane on the left, click **Instance Management**.


**Step 3** Click **Download Client Configuration** on an instance card.

**Step 4** Select configuration files to download.



SDK Configuration File 


\* Chaincode Name  

\* Certificate Path  


\* Channel

\* Peer


---

Orderer Certificate 

---

Peer Certificates 

\* Peer Organization

Certificate   

- **SDK Configuration File:** Specify the member, chaincode name, certificate path as required.

**Table 2-17** SDK file parameters

Parameter	Description
Chaincode Name	Set it as required. The chaincode name must be the same as the name specified during chaincode installation and instantiation.

Parameter	Description
Certificate Path	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.
Channel	Select a channel.
Peer	Select peer organizations in the channel.

- An orderer certificate is used for interacting with the blockchain system. Encrypt the private keys in the downloaded certificates for storage.
- A peer certificate is used for performing management operations within a channel. Encrypt the private keys in the downloaded certificates for storage. Select a peer organization and the certificates to be downloaded.

**Step 5** Click **Download**. Decompress the SDK and store the retrieved .yaml file. Decompress the downloaded certificate packages and store the files in an application directory for the application to access.

----End

## 2.8 Consortium Management

### 2.8.1 Forming a Consortium

After creating a consortium blockchain, you can invite tenants to join it. In addition, you can invite others through different channels to form a consortium blockchain.

#### Inviting a Tenant

Create a consortium blockchain to invite others to join the consortium.

**Step 1** Log in to the BCS console.

**Step 2** Click **Member Management** in the navigation pane on the left. Click **Invite Tenant** in the upper right corner of the page.

**Step 3** In the **Invite Tenant** window, select your BCS instance and channel, and enter the invitee's name.

**Figure 2-40** Inviting a tenant

Invite Tenant ?

! Ensure that the account name is correct. You can check the account name on the [My Credentials](#) page.

Service  Consortium Channel

\* Invitee

+ Add Tenant

**Step 4** (Optional) Click **Add Tenant** to invite multiple tenants.

 **NOTE**

A maximum of 40 tenants can be invited.

**Step 5** Click **OK**. An invitation notification is sent to the invitee.

----End

## Accepting/Declining an Invitation

When you are invited to join a consortium blockchain, you will receive a notification. You can either accept or decline it.

**Step 1** Log in to the BCS console.

**Step 2** Click **Notification Management** in the navigation pane on the left. On the **Notification Management** page, locate the notification and click **View Details** in the **Operation** column.

- To accept the invitation, select the organization that you want to add to the consortium, and then click **Accept**.
- To decline the invitation, click **Decline**.

 **NOTE**

- An invitee can select an existing BCS instance from the drop-down list box or click **Create Instance** to create a new one.  
An invitee can accept invitations sent by only one inviting party. To accept invitations from other inviting parties, the invitee must create new BCS instances.  
If an invitee receives multiple invitations from multiple channels of an inviting party, the invitee can create a BCS instance using one of the channels, and use the same BCS instance to accept invitations from other channels.
- For details about how to create a BCS instance, see [Instance Deployment](#). To successfully join a consortium blockchain, certain parameters of your instance must have the same settings as the inviting party's BCS instance, such as the blockchain type, consensus mechanism, and security mechanism. Therefore, these parameters are dimmed on the instance configuration page and cannot be modified.

----End

## 2.8.2 Member Management

You can invite tenants to become blockchain consortium members, who can view invitations and topologies and delete invitations.

- To invite a tenant, click **Invite Tenant** in the upper right corner of the **Member Management** page. For details, see [Inviting a Tenant](#).
- To view an invitation, click **View Invitation** in the **Operation** column on the **Member Management** page.
- To delete an invitation, click **Delete Invitation** in the **Operation** column on the **Member Management** page. After you delete an invitation, it is withdrawn. This operation can be done only if the invitee has not accepted the invitation.
- To view the topology between consortium blockchain members, click **View Topology** in the **Operation** column on the **Member Management** page.

You can invite a tenant to join a channel to establish a consortium blockchain. Tenants cannot be invited to a private blockchain.

## 2.8.3 Notification Management

When another tenant invites you to join a consortium blockchain, you will receive an invitation notification. Then, you can view the invitation on the **Notification Management** page.

- To accept the invitation, click **View Details** in the **Operation** column of the notification list, select a BCS instance and organization, and click **Accept**.
- To decline the invitation, click **View Details** in the **Operation** column of the notification list, and click **Decline**.
- To delete a notification, click **Delete Notification** in the **Operation** column of the notification list
- To postpone the processing of an invitation, click **View Details** in the **Operation** column of the notification list, and click **Process Later**.

### NOTE

- Click **Create Instance** and use the new BCS instance to join the channel.
- Notification statuses include:
  - **Unprocessed**: You have not processed the invitation notification. You can click **View Details** to accept or decline the invitation.
  - **Finished**: You have accepted the invitation to join the consortium blockchain.
  - **Canceled**: The inviting party has deleted the instance before you accept the invitation. You cannot join the consortium blockchain.
  - **Declined**: You have declined the invitation to join the consortium blockchain.
  - **Quit**: You have accepted the invitation and joined the consortium blockchain but later quit the consortium.
  - **Dismissed**: The inviting party has deleted the instance after you joined the consortium blockchain. As a result, the blockchain is dismissed.
  - **Upgraded**: An instance in the consortium blockchain has been upgraded after you join the blockchain.

## 2.9 Add-on Management

### 2.9.1 Add-on Overview

Add-ons allow you to extend the functionality of BCS instances as required. On the **Add-on Management** page, you can install add-ons and upgrade, uninstall, and view details about the installed add-ons. [Table 2-18](#) shows the add-ons.

**Table 2-18** Add-ons

Name	Description	Restrictions
baas-restapi	Supports access to the blockchain system by using RESTful APIs. Supports management capabilities such as generation, application, and issuance of distributed identities and verifiable credentials, as well as data release, authorization, sharing, decryption.	<p>This add-on can be installed only if the BCS instance meets all of the following conditions:</p> <ul style="list-style-type: none"> <li>Enhanced Hyperledger Fabric architecture</li> <li>Deployed in a CCE cluster</li> <li>v3.0.16 or later (corresponding to Hyperledger Fabric v1.4.0) or v4.0.5 or later (corresponding to Hyperledger Fabric v2.2)</li> <li>Endorsement is from any organization under the BCS instance</li> <li>Uses ECDSA for the security mechanism</li> </ul>

## Installing the baas-restapi Add-on

- Step 1** Log in to the BCS console.
- Step 2** Click **Add-on Management** in the navigation pane on the left.
- Step 3** On the **Add-on Repository** tab page, click **Install** on the card of the **baas-restapi** add-on.
- Step 4** Set the parameters by referring to [Table 2-19](#).

**Table 2-19** Parameters

Parameter	Description	Example Setting
Add-on	Add-on name.	baas-restapi
Version	Add-on version.	3.0.26
Instance	Select a BCS instance.	bcs-6zbgus
Enable DID API	<p>Allows you to manage DIDs, generate, apply, issue verifiable credentials.</p> <p>Determine whether to enable the distributed identity APIs based on the service requirements.</p>	-

Parameter	Description	Example Setting
Enable APIs for Trusted Data Exchange	Allows you to publish, authorize, share, and decode data. Determine whether to enable the trusted data exchange APIs based on the service requirements. <b>NOTE</b> This parameter is displayed only when <b>Enable DID API</b> is enabled.	-
Channel	Select a channel for installing chaincode. <b>NOTE</b> This parameter is displayed only when <b>Enable DID API</b> is enabled.	channel

**Step 5** Click **Next**.

 **NOTE**

Do not perform operations on the instance when installing an add-on.

----End

## Add-on Instances

**Step 1** Log in to the BCS console.

**Step 2** Click **Add-on Management** in the navigation pane on the left.

**Step 3** View the add-ons on the **Add-on Instances** tab page.

You can perform the following operations on the add-ons as required:

- **baas-restapi:**
  - Click the add-on to view its details.
    - You can click **Scale** next to **Normal/All Instances** to scale the number of instances in the range from 1 to 5.
    - Click **Modify** to enable or disable the APIs for DID and trusted data exchange. After you click **OK**, the instance will be restarted and will be interrupted for a short period of time. Refresh the page later.
  - Click **Uninstall** to uninstall an add-on.

----End

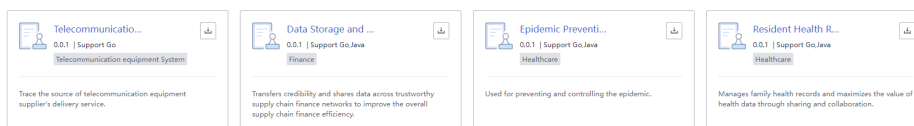
## 2.10 Contract Repository

A contract template is a smart contract that can implement certain functions. You can directly use the code provided by the templates or use the templates as a foundation for developing your own smart contracts.

In the Contract Management module on the console, you can view contract templates for various industries, download the ones you need, and manage your contract templates.

## Downloading a Contract Template

- Step 1** Log in to the BCS console.
- Step 2** Click **Contract Repository** in the navigation pane on the left.
- Step 3** On the **Contract Repository** tab page, view contract templates for different industries, such as finance, healthcare, energy, and aviation.



- Step 4** Click the template name to view details about a contract template, including the version, supported language, category, and interfaces.

**Figure 2-41** Viewing contract details

**Contract Details**
×

---

**Data Storage and Query**

Version: 0.0.1

Language: Go,Java

Category: Finance

Description: This contract template facilitates sharing of data such as accounts receivable/payable and key contracts across blockchain consortium-based supply chain systems. It allows the credibility of supply chain participants to be transferred based on the endorsement of core enterprises.

**Interfaces**

Interface	Parameter	Description
saveRecord	<a href="#">View Details</a>	Saves records
queryRecord	<a href="#">View Details</a>	Queries records
queryRecordByPartial	<a href="#">View Details</a>	Queries records by key...
deleteRecord	<a href="#">View Details</a>	Delete records
setKeyType	<a href="#">View Details</a>	Sets the key type
getKeyType	<a href="#">View Details</a>	Queries the key type

- Step 5** Click to download a contract template.

You can use the downloaded template files to install and instantiate chaincodes. For details, see [Chaincode Management](#).

----End

# 3 FAQs

---

## 3.1 BCS FAQs

### 3.1.1 Instance Management

#### 3.1.1.1 Consultation

##### 3.1.1.1.1 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.

##### 3.1.1.1.2 What Underlying Framework Is Used for BCS?

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.

### 3.1.1.1.3 What Competitive Advantages Does 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).

### 3.1.1.1.4 What Are the Specifications of VMs to Be Created for BCS?

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

**Table 3-1** Suggestions on VM creation

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.

### 3.1.1.1.5 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.

### 3.1.1.1.6 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 3-2** 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

### 3.1.1.1.7 When Do I Need to Hibernate or Wake an Instance?

#### Scenario

You can hibernate a BCS instance when it is not required temporarily and wake it when you need to use it. Instances in hibernation are unavailable.

### 3.1.1.2 Service Usage

#### 3.1.1.2.1 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.
- Step 2** Click an instance name on the **Instance Management** page. 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

#### 3.1.1.2.2 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](#).

### 3.1.1.2.3 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.

### 3.1.1.2.4 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.

### 3.1.1.2.5 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.

### 3.1.1.2.6 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.

## This site is not secure

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

 [Close this tab](#)

 [More information](#)

**Your PC doesn't trust this website's security certificate. The hostname in the website's security certificate differs from the website you are trying to visit.**

Error Code: DLG\_FLAGS\_INVALID\_CA  
DLG\_FLAGS\_SEC\_CERT\_CN\_INVALID

 [Go on to the webpage \(not recommended\)](#)

- 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.



## Your connection is not private

Attackers might be trying to steal your information from  .36 (for example, passwords, messages or credit cards). [Learn more](#)

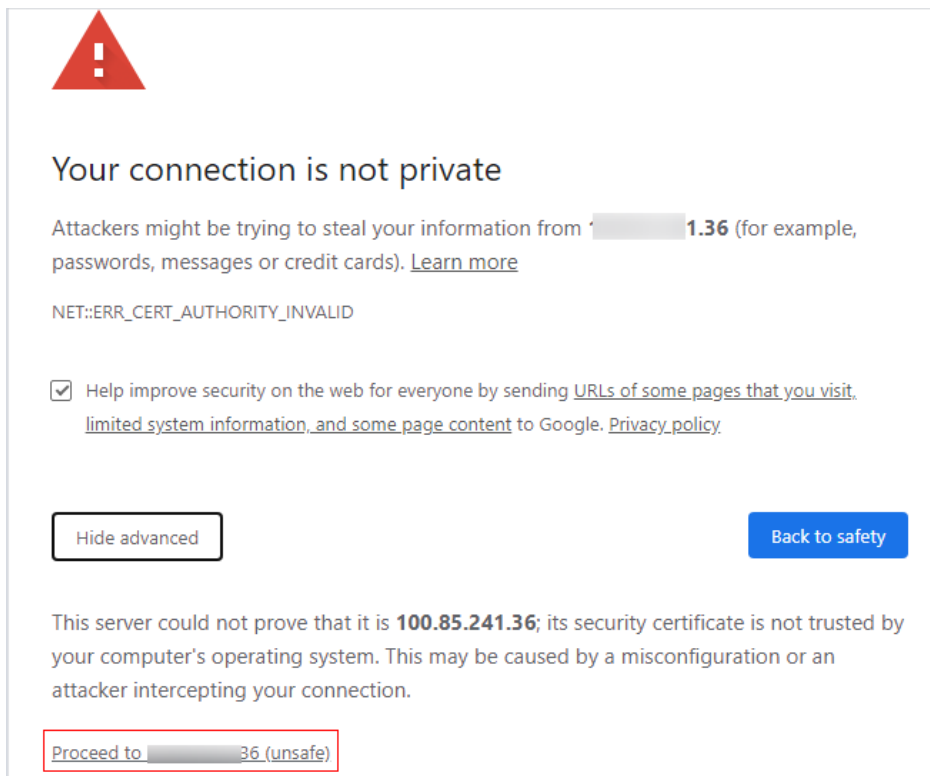
NET::ERR\_CERT\_AUTHORITY\_INVALID

Help improve security on the web for everyone by sending [URLs of some pages that you visit, limited system information, and some page content](#) to Google. [Privacy policy](#)

[Advanced](#)

[Back to safety](#)

- b. Click **Advanced**.



- c. Proceed to the login page.

### 3.1.1.2.7 BCS.4009100: System Error

#### Symptom


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

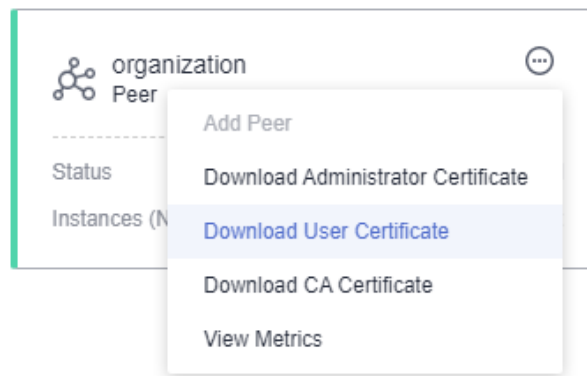
#### Solutions for Common Scenarios

- **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.

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

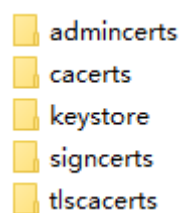
Download the private keys and certificates on the BCS console or generate them using OpenSSL.


- 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 the **Enhanced Hyperledger Fabric** tab and click an instance to view its details.
  - c. In the **Blockchain Organizations** area, click  to download the user certificate.

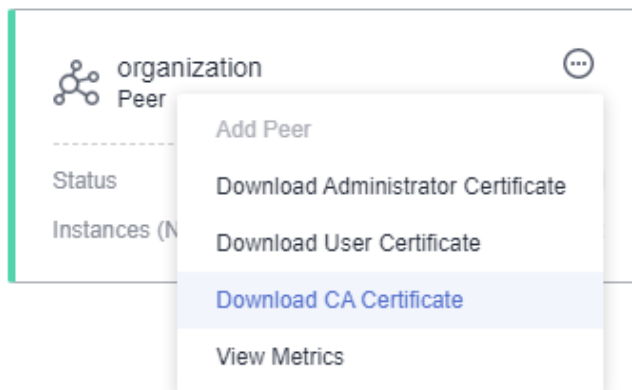


- d. Decompress the downloaded user certificate. The **msp** folder contains the user private key (**keystore**) and certificate (**signcerts**), as shown in the following figure.

**Figure 3-1** File directory

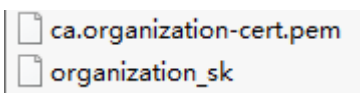


- To generate the private keys and certificates of multiple users, use OpenSSL.
  - a. Download the CA certificates and private keys.
    - i. Log in to the BCS console.
    - ii. In the navigation pane on the left, click **Instance Management**. Click the **Enhanced Hyperledger Fabric** tab and click an instance to view 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 3-2** 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**.

### 3.1.1.2.9 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, and choose **Resource Management > Storage** in the navigation pane.
  - d. On the **SFS Turbo** tab page, select the target BCS instance's cluster, and click **Expand Capacity** in the row containing the recorded PVC.
- Contact technical support to back up data.

### 3.1.1.3 Abnormal Instance Statuses

### 3.1.1.3.1 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, choose **Resource Management > 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, choose **Resource Management > 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, choose **Resource Management > Storage**. Click **SFS Turbo** and check the status of the file system in the CCE cluster where the abnormal BCS blockchain is deployed.
    - iii. If the SFS Turbo volume is in the **Lost** state, the state of the BCS instance will be displayed as **Abnormal**.

**Figure 3-3** Abnormal PVC

The screenshot shows the SFS Turbo console interface. At the top, there are navigation tabs: EVS, SFS, OBS, SFS Turbo (selected), and Snapshot and backup. Below the tabs, there is a warning message: "Creation of SFS Turbo file systems is temporarily not supported. Go to the SFS console and create an SFS Turbo file system in VPC vpc-bcs-zhwa and subnet subnet-bcs-9g9m. Make sure that the security group of the SFS Turbo file system to be created have ports 111, 445, 2049, 2051, 2052, and 2048 opened." Below the warning, there is a search bar and a table of PVCs. The table has columns: PVC Name, Volume Name, PVC Status, Total Capa..., Endpoint, Protocol, Type, Namespace, Created, Encrypted, Storage Fo..., and Operation. The first row in the table is highlighted, and the 'PVC Status' column for this row is circled in red, showing the status 'Lost'.

PVC Name	Volume Name	PVC Status	Total Capa...	Endpoint	Protocol	Type	Namespace	Created	Encrypted	Storage Fo...	Operation
cce-efs-import-aaa38kem-85f4	N/A	Lost	500	N/A	N/A	Standard	default	Jun 11, 20...	No	CSI	Unbind   Expand Capacity



**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 3-4** top command details

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7356	root	20	0	1219932	37648	37144	S	1.7	1.2	06:03.44	kubelet
10635	1000	20	0	336628	117752	17732	S	1.3	1.5	50:17.23	peer
6437	root	20	0	4292872	101804	29996	S	1.0	1.3	59:15.45	dockerd
7324	root	10	-10	530536	66036	5468	S	0.7	0.0	22:24.03	ovs-vsuidhd
8189	root	20	0	592912	37340	17392	S	0.7	0.5	8:42.00	canal-agent
10630	root	20	0	657536	41664	16592	S	0.7	0.5	9:21.68	everest-csi-con
12216	root	20	0	251064	38328	8924	S	0.7	0.5	25:05.20	icagent
4583	root	20	0	4907740	116272	13264	S	0.3	1.5	2:09.03	java
5655	root	20	0	123920	21500	7536	S	0.3	0.3	0:27.93	daeventd
5900	root	20	0	139936	30000	13900	S	0.3	0.4	6:59.20	kube-proxy
6115	root	20	0	113264	1620	1352	S	0.3	0.0	0:51.39	srvcubeproxy
10497	1000	20	0	265076	51396	10012	S	0.3	0.6	10:07.41	baas-agent
1	root	20	0	191812	4804	2524	S	0.0	0.1	10:16.63	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.05	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:07.20	kssoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworke/0:0H
7	root	rt	0	0	0	0	S	0.0	0.0	0:03.92	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	2:19.75	rcu_sched
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.70	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.69	watchdog/1
13	root	rt	0	0	0	0	S	0.0	0.0	0:27.90	migration/1
14	root	20	0	0	0	0	S	0.0	0.0	0:10.60	kssoftirqd/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:02.41	kworke/1:0H
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.50	watchdog/2
18	root	rt	0	0	0	0	S	0.0	0.0	0:03.01	migration/2
19	root	20	0	0	0	0	S	0.0	0.0	0:12.10	kssoftirqd/2
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworke/2:0H
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.61	watchdog/3
23	root	rt	0	0	0	0	S	0.0	0.0	0:27.67	migration/3
24	root	20	0	0	0	0	S	0.0	0.0	0:10.03	kssoftirqd/3
26	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworke/3:0H
27	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
28	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
29	root	20	0	0	0	0	S	0.0	0.0	0:00.22	khungtaskd
30	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	wakeback
31	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
32	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
33	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
34	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset

- 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.

### 3.1.1.3.2 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, choose **Resource Management > 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. To wake up a cluster, choose **More > Wake** for the target cluster.
- 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, choose **Resource Management > 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.

### 3.1.1.3.3 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 view the EIP.
2. Go to the Network Console, locate the target EIP, and view 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, create a new EIP and bind it. For details, see [Assigning an EIP and Binding It to an ECS](#). 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**.

- 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.

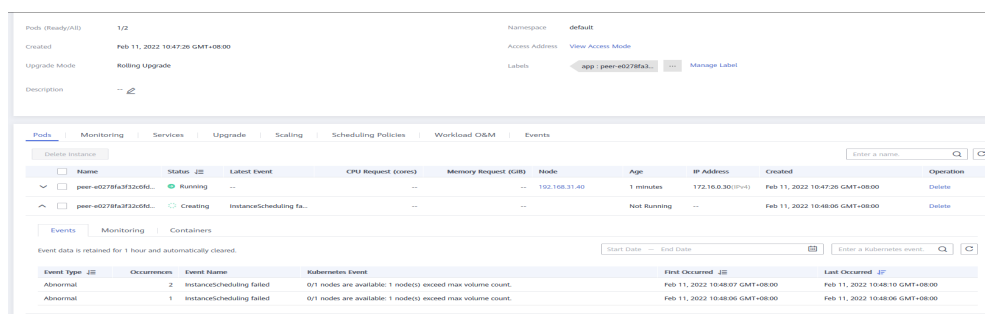
Solution: Log in to the ECS console, click a target ECS, on the **Tags** tab page, add a tag to the ECS instance. Set the tag key to **CCE-Dynamic-Provisioning-Node** and the tag value to any number.

### 3.1.1.3.4 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 3-5 Abnormal pods



#### Solution

- 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.

- Choose **Resource Management > 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, view the pod list and check whether the pods are normal.

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

----End

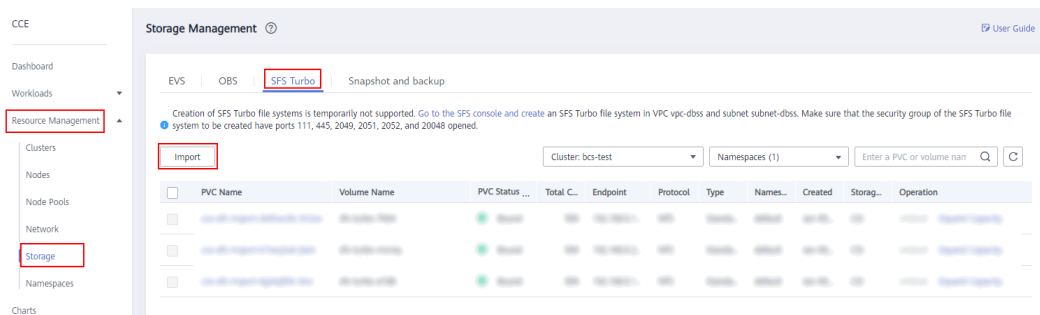
### 3.1.1.4 Other Issues

#### 3.1.1.4.1 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 creating 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 creation. 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.

### 3.1.1.4.2 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}
```

## 3.1.2 Chaincode Management

### 3.1.2.1 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.

### 3.1.2.2 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

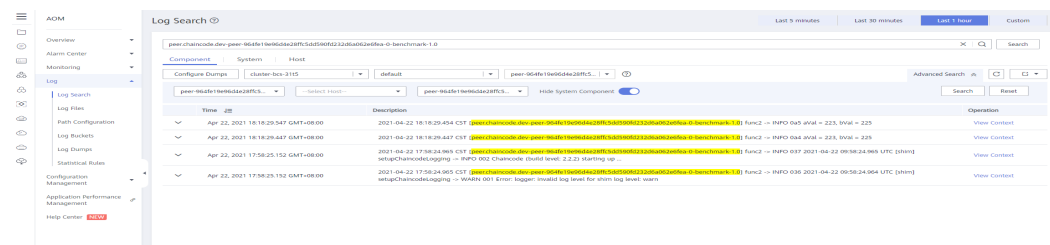
of chaincode containers. On the AOM console, only the log files of the chaincode containers started using Kubernetes are displayed. Therefore, you cannot view the log files of chaincode containers started using Docker.

## 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 3-6 Chaincode logs



### 3.1.2.3 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.

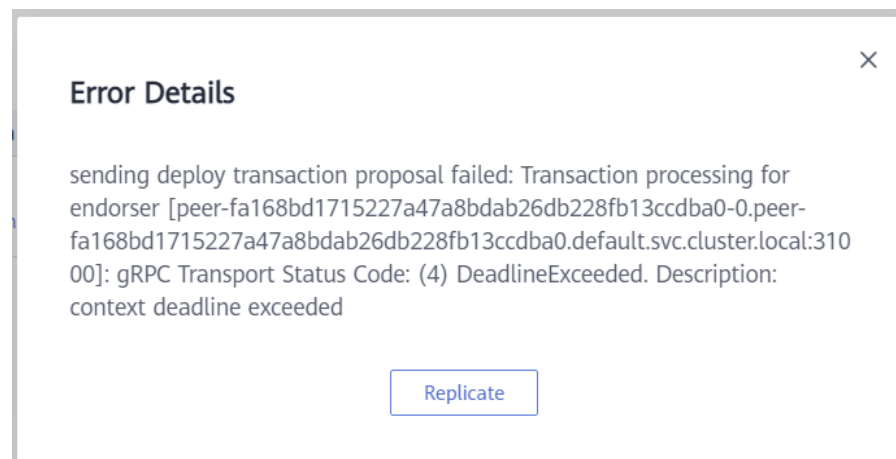
#### Solution

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

### 3.1.2.4 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 3-7** 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.

## 3.1.3 Data Storage to the Blockchain

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

#### Symptom

Transaction connections fail or time out.

#### Fault Locating

**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 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.

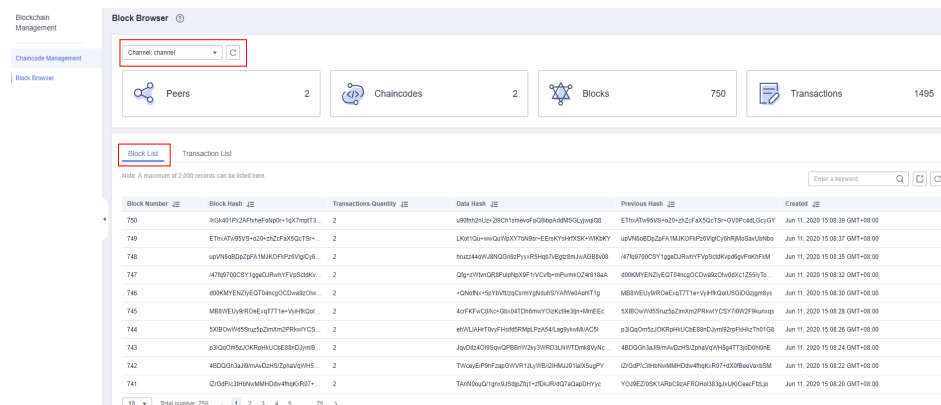
### Solution

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

Download link: <https://github.com/hyperledger/fabric>

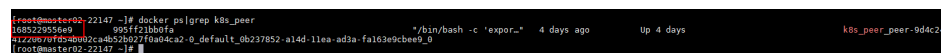
- Check item 4: Check whether the ledger of the peer is updated.
  - Log in to the BCS console. In the navigation pane, click **Instance Management**. On the card containing the abnormal instance, click **Manage Blockchain**. On the **Block Browser** page, select the abnormal channel, and view the block quantity on the **Block List** tab page.

Figure 3-8 Block List page



- 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 3-9 Checking the peer containers



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

Figure 3-10 Accessing the container





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

Figure 3-11 Querying the channel to which the peer is added

```
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$ peer channel list
2020/06/02 11:41:44 proto: duplicate proto type registered: msp.VersionedValueProto
2020-06-02 11:41:44.088 CST [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2020-06-02 11:41:44.110 CST [main] SetOrdererEnv -> WARN 002 CORE_LOGGING_LEVEL is no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2020-06-02 11:41:44.126 CST [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
Channels peers has joined
channel
```

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

Figure 3-12 Checking whether the ledger of the peer is updated

```
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$ peer channel getinfo -c channel
2020/06/02 11:42:32 proto: duplicate proto type registered: msp.VersionedValueProto
2020-06-02 11:42:32.157 CST [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2020-06-02 11:42:32.187 CST [main] SetOrdererEnv -> WARN 002 CORE_LOGGING_LEVEL is no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2020-06-02 11:42:32.204 CST [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
Blockchain info: {"height":11,"currentBlockHash":"121bc5FRnFwq1/FNPL187CKXHEdWnDyCd9Q3UBRU=","previousBlockHash":"UX1fm5JPYl8idsjVc51gBYHU+nVdXkj+uQA/LAw="}
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$
```

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.

- 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 3-13 Checking the peer containers

```
[root@luster-bcs-rvjo-jhy1 ~]# docker ps | grep k8s_peer
42538159755c          a7ecd6c6549          "/bin/bash -c 'expor..." 22 hours ago      Up 42 minutes      k8s_peer_peer
aa006408de5b53911271bc3d8ce77d32c074939e-0_default_6741801f-ai4d-4c96-85a1-d85538fa981_0
```

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

Figure 3-14 Accessing the container

```
[root@luster-bcs-rvjo-jhy1 ~]# docker ps | grep k8s_peer
42538159755c          a7ecd6c6549          "/bin/bash -c 'expor..." 22 hours ago      Up 42 minutes      k8s_peer_peer
aa006408de5b53911271bc3d8ce77d32c074939e-0_default_6741801f-ai4d-4c96-85a1-d85538fa981_0
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$
```

- 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.

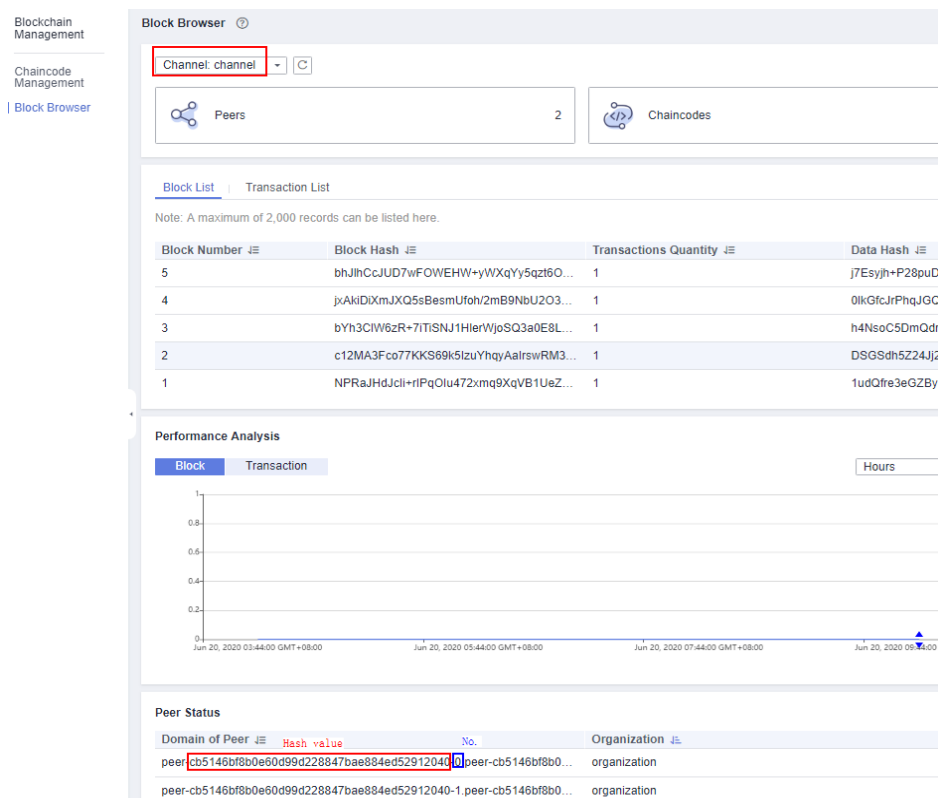
Figure 3-15 Viewing all files

```
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 fabric]$ cd /var/log/baas-service/peer/
[paas@peer-aa006408de5b53911271bc3d8ce77d32c074939e-0 peer]$ ll
total 816
-rw-r----- 1 paas paas 909 Jun 1 21:18 audit_peer-aa006408de5b53911271bc3d8ce77d32c074939e-0.log
-rw-r----- 1 paas paas 0 May 29 11:04 audit_peer-c9b76a60ef8db9e85f21692b6053fea246ef5e6-0.log
-rw-r----- 1 paas paas 233691 Jun 2 16:29 check.log
-rw-r----- 1 paas paas 344671 Jun 2 15:47 peer-aa006408de5b53911271bc3d8ce77d32c074939e-0.trace
-rw-r----- 1 paas paas 239941 May 29 16:07 peer-c9b76a60ef8db9e85f21692b6053fea246ef5e6-0.trace
```

- 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 3-16 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.

- 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 CloudDB is used for ledger storage for an instance of an earlier version, the status data is not stored in the SFS file system. If BCS is restarted, CouchDB will reload the block data to generate the status data, and BCS will be 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.
  - 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.

### 3.1.3.2 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.

### 3.1.3.3 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](#).

### 3.1.3.4 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.

## 3.1.4 Demos and APIs

### 3.1.4.1 Demo Problems

### 3.1.4.1.1 General Checks

## Fault Locating

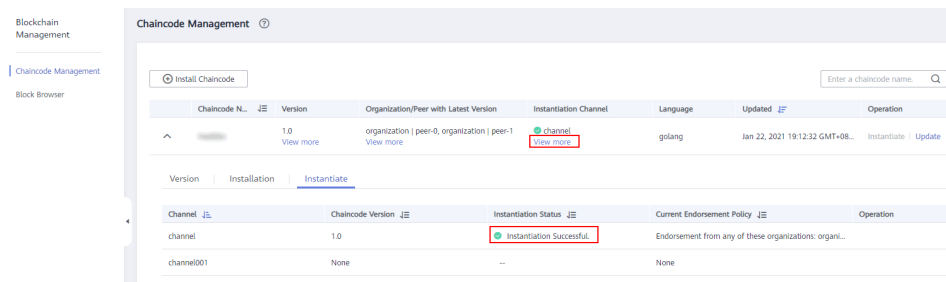
**Check item 1: Check whether the instance status is normal.**

**Check item 2: Check whether the chaincode is properly instantiated.**

**Check item 3: Check whether the demo has been modified.**

## Solution

- Check item 1: 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 2: 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 3: Check whether the demo has been modified.  
Follow the instructions in the next sections to check whether the specified demo has been modified.

## 3.1.5 O&M and Monitoring

### 3.1.5.1 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
```

### **3.1.5.2 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.

## **3.1.6 Consortium Management**

### **3.1.6.1 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.

# 4 Change History

---

Release On	Change History
2023-04-30	Optimized descriptions in <a href="#">Edition Differences</a> .
2022-12-30	Optimized descriptions.
2021-01-15	First official release