Alibaba Cloud Container Service

Block Chain

Issue: 20180808

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade

secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 Version history and upgrade instructions

This document records the version history and the corresponding function changes of the Alibaba Cloud Container Service blockchain solution, and the instructions when you upgrade the blockchain solution. The version number is the chart version of the blockchain solution ack-hyperledger-fabric in the app catalog.

Version history

Version 0.2.1

- Chart is renamed as ack-hyperledger-fabric.
- Bug fixes.

Version 0.2.0

- Supports Hyperledger Fabric 1.1.0.
  - Supports the Node.js-type chaincode function and the related examples.
  - Supports the chaincode-level ledger data encryption function and the related examples.
  - Supports connection profile.
  - Code-level optimization to further improve the performance and horizontal scalability.
  - Other new functions provided by Hyperledger Fabric 1.1.0.
- Integrates with Alibaba Cloud Log Service.
- Built-in deployment of Hyperledger Blockchain Explorer.
- Supports Alibaba Cloud Elastic Compute Service (ECS) Bare Metal Instance.
- Optimizes the cleanup of the data directory in the process of deleting the blockchain network.

Version 0.1.0

- Supports deploying Hyperledger Fabric 1.0.0 on an Alibaba Cloud Container Service Kubernetes cluster.

Upgrade instructions

- The blockchain solution installed on the App Catalog page in the Container Service console is of the latest version.

- For the blockchain solution installed by using the **`helm install`** command, use the following command to view the current version because helm repo may have the cache of an earlier version locally if the blockchain solution was installed before.

  ```
  helm search hyperledger
  ```

  To update the local repo cache, run the following command to obtain the blockchain solution of the latest version:

  ```
  helm repo update
  ```

2 Overview

Alibaba Cloud Container Service blockchain solution is a simple and flexible generic solution used to develop and test the blockchain applications and solutions based on *Hyperledger Fabric*.

This solution provides developers with a graphical blockchain network configuration wizard to address the issues in blockchain configuration and deployment, such as high requirements for professional skills, complex and time-consuming procedures, and being prone to errors. By using this solution, developers are only required to enter key configuration parameters. Then , with the one-click automated configuration and deployment, the complex configuration files can be generated and the blockchain network based on Hyperledger Fabric can be created in the container cluster within several minutes. In addition, to facilitate developers to access the blockchain network by using blockchain applications and blockchain administration and monitoring tools, the solution provides pre-generated configuration files for applications and tools . Developers can download the configuration files with one click and then use the blockchain applications and tools.

Alibaba Cloud Container Service blockchain solution has the following characteristics:

- Standard: Supports major functions of the open-source blockchain technology Hyperledger Fabric V1.1 in the Linux foundation, including standard blockchain node types such as Peer, CouchDB, Orderer, Kafka, ZooKeeper, and CA.
- Simple: A graphical wizard that simplifies blockchain configuration, masks complicated underlying procedures such as parameter settings, tool invocation, and configuration distribution, significantly reduces the error probability, and supports the built-in deployment of Hyperledger Blockchain Explorer.
- Mature: Multiple years of enterprise level large-scale production and application have proved the stability and reliability of underlying Alibaba Cloud Container Service clusters. In addition, Alibaba Cloud Container Service clusters integrate with best practices of cloud applications and expertise of Docker technology development.
- Versatile: Seamlessly integrates with powerful and rich enterprise level application service capabilities of Alibaba Cloud, such as storage, network, routing, security, image, log, and monitoring, and provides all-round support for blockchain upper-layer applications. In the future , the blockchain solution will support multiple deployment modes such as public could, private cloud, and hybrid cloud.

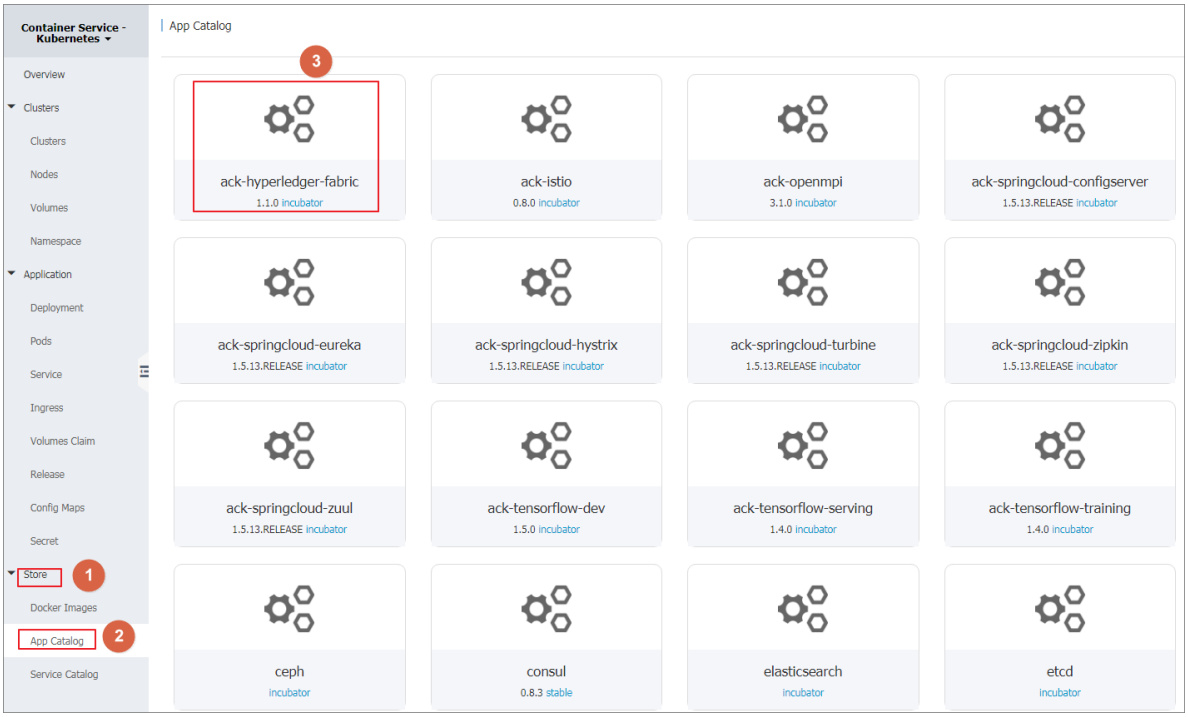
Alibaba Cloud Container Service blockchain solution is currently in public beta. Welcome to try the solution and provide your valuable comments and suggestions if you are interested in it. This solution is to be improved and enriched continuously according to your requirements and the market requirements, allowing developers and partners to create more and better service innovation applications and industry solutions.

5 Access blockchain solution homepage

Alibaba Cloud Container Service blockchain solution is released in the Container Service Kubernetes app catalog in the form of Helm chart. You can follow these steps to view the blockchain solution information, such as the introduction and configuration parameters.

Procedure

1. Log on to the *Container Service console*.
2. Click Kubernetes > Store > **App Catalog** in the left-side navigation pane.
3. Click **ack-hyperledger-fabric** on the App Catalog page.



4. On the details page of ack-hyperledger-fabric, view the blockchain solution information, including the introduction, deployment conditions, deployment commands, test commands, debugging commands, and configuration parameters.

6 Configure and deploy blockchain network

After completing the environment preparations, you must configure and deploy the blockchain network. Fabric and consisted of the following standard node types: The blockchain network is a blockchain running environment, which is based on Hyperledger Fabric and consisted of the following standard node types:

- Orderer: Used to combine blockchain transactions into blocks. From the perspective of scalability, the blockchain solution uses the Orderer services of the Kafka type.
- Kafka and ZooKeeper: Used to provide underlying services for Orderer in the form of clusters.
- Peer: Used to store and maintain a ledger, create and run a chaincode container, and endorse transactions. From the perspective of high availability, the blockchain solution creates two Peer nodes for each organization.
- CouchDB: Used to store the state database of Peer. The blockchain solution creates a CouchDB for each Peer.
- CA: Used to provide the PKI certificate service for applications. The blockchain solution creates a CA node for each organization.

For more information, see *Hyperledger Fabric* official document.

To satisfy requirements of enterprise-level applications, the blockchain solution provides master nodes with persistent data storage by using the shared file system created in ##### Environment preparations.

The blockchain network is deployed and run on an Alibaba Cloud Container Service Kubernetes cluster. You can deploy multiple blockchain networks (separated by namespaces) in the same Kubernetes cluster or deploy one blockchain network in each Kubernetes cluster.

The blockchain network can be configured and deployed on GUI or by using the Helm commands . Both of these methods are introduced in the following sections. Both of these methods are introduced in the following sections.

Note:

Currently, the new organizations or Peers cannot be dynamically added to an existing blockchain network. Therefore, you must delete the existing blockchain network and recreate one if you want to modify the blockchain network configurations.

Configuration parameter description

The blockchain solution provides default values for most parameters to simplify your configurations. To customize the blockchain, you can see the following parameter descriptions and then configure the parameters.

Parameter	Description
dockerImageRegistry	The Docker image repository URL: The image repository of the blockchain solution. registry.cn-beijing.aliyuncs.com/cos-solution - China North 3 (Zhangjiakou): registry.cn-zhangjiakou.aliyuncs.com/cos-solution - China East 1 (Hangzhou): registry.cn-hangzhou.aliyuncs.com/cos-solution - China East 2 (Shanghai): registry.cn-shanghai.aliyuncs.com/cos-solution - China South 1 (Shenzhen): registry.cn-shenzhen.aliyuncs.com/cos-solution Select an image repository based on the region in which the Kubernetes cluster resides. If you use an image repository in another region, the Internet traffic cost may be generated when you deploy the blockchain network. - China North 1 (Qingdao): registry.cn-qingdao.aliyuncs.com/cos-solution - China North 2 (Beijing): registry.cn-beijing.aliyuncs.com/cos-solution - China North 3 (Zhangjiakou): registry.cn-zhangjiakou.aliyuncs.com/cos-solution - China East 1 (Hangzhou): registry.cn-hangzhou.aliyuncs.com/cos-solution - China East 2 (Shanghai): registry.cn-shanghai.aliyuncs.com/cos-solution - China South 1 (Shenzhen): registry.cn-shenzhen.aliyuncs.com/cos-solution
fabricNetwork	Blockchain network name (required): The blockchain network is deployed as an application of Container Service. Therefore, the blockchain network name is an application name. Avoid using the name of a deployed application. The blockchain network name is also used as the name of the root directory for storing configurations and data in the shared file system.

Parameter	Description
fabricChannel	Blockchain network channel name: The channel name of Hyperledger Fabric. The blockchain solution automatically creates the channel with the specified name when the blockchain network is deployed.
externalAddress	External address (required): To access the blockchain network by using an application or administration and monitoring tool that is deployed outside the container cluster, you must provide the Internet address of a node in the Kubernetes cluster in which the blockchain network is deployed or the Internet address of the Server Load Balancer instance of the Kubernetes cluster as the external address. Bind an EIP to a worker node in Environment preparations. For more information about the configuration method, see #### Bind an EIP to a worker node in Environment preparations.
ordererDomain	The Orderer domain in Hyperledger Fabric. Set this parameter based on actual requiremen ts.
ordererNum	Number of Orderer nodes: This parameter applies to the Orderer services of the Kafka type, not the Solo type. Set this parameter to the number of Orderer nodes to be deployed . To modify the parameter value, you must modify the value of ordererExternalPortList at the same time to make sure the number of nodes and the number of external ports are the same. Otherwise, the blockchain network fails to be deployed. nodes to be deployed. To modify the parameter value, you must modify the value of ordererExternalPortList at the same time to make sure the number of nodes and the number of external ports are the same . Otherwise, the blockchain network fails to be deployed.
peerDomain	Peer domain: The Peer domain in Hyperledge r Fabric. Set this parameter based on actual requirements.

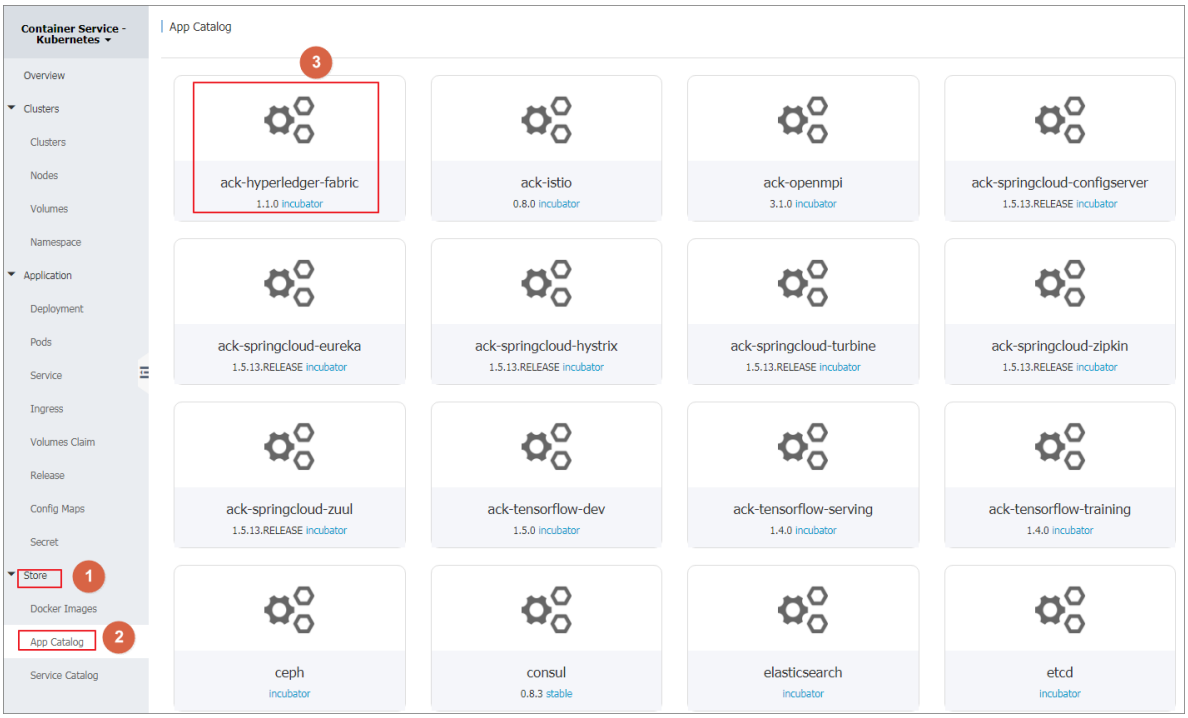
Parameter	Description
orgNum	Number of organizations: The number of organizations in Hyperledger Fabric. The blockchain solution creates two Peer nodes for each organization to guarantee the high availability and meet the requirements of business expansion. Specify the number of organizations according to your actual requirements. The number of actually deployed Peer nodes is twice the number of organizations. To modify the parameter value, you must modify the values of peerExternalGrpcPortList, peerExternalEventPortList, and caExternalPortList at the same time to make sure the number of nodes and the number of external ports are the same. Otherwise, the blockchain network fails to be deployed. caExternalPortList at the same time to make sure the number of nodes and the number of external ports are the same. Otherwise, the blockchain network fails to be deployed.
ordererExternalPortList	Orderer external port list: To access the Orderer services by using an application outside the cluster, you must specify the external ports used by Orderer nodes or use the default port. Different blockchain networks must use different ports and make sure the number of ports in the list must be the same as the value of ordererNum. Otherwise, the blockchain network fails to be deployed. ordererNum. Otherwise, the blockchain network fails to be deployed.
caExternalPortList	CA external port list: To access the CA services by using an application outside the cluster, you must specify the external ports used by CA nodes or use the default port. Different blockchain networks must use different ports and make sure the number of ports in the list must be the same as the value of orgNum. Otherwise, the blockchain network fails to be deployed.

Parameter	Description
peerExternalGrpcPortList	Peer gRPC external port list: To access the Peer services by using an application outside the cluster, which is based on the gRPC protocol by default, you must specify the external ports used by Peer nodes or use the default port. Different blockchain networks must use different ports and make sure the number of ports in the list must be twice the value of orgNum. Otherwise, the blockchain network fails to be deployed.
peerExternalEventPortList	Peer event external port list: To access the Peer event services by using an applicatio n outside the cluster, you must specify the external ports used by Peer nodes or use the default port. Different blockchain networks must use different ports and make sure the number of ports in the list must be twice the value of orgNum. Otherwise, the blockchain network fails to be deployed.
imagePullPolicy	Image pull policy: This is a Kubernetes parameter and is generally used for development and test.
hyperledgerFabricVersion	Hyperledger Fabric version: Currently, 1.1.0 is supported, and no configuration is required.
thirdPartyImageVersion	The image version of the third-party softwares (such as CouchDB, Kafka, and ZooKeeper) included in Hyperledger Fabric. Currently, 0 .4.6 is supported, which corresponds to the Hyperledger Fabric 1.1.0, and no configuration is required. Currently, 0.4.6 is supported, which corresponds to the Hyperledger Fabric 1.1.0, and no configuration is required.
explorer.enabled	Whether or not to automatically deploy the Hyperledger Explorer. During the deployment, the Server Load Balancer instance is created, and the blockchain explorer function based on Web UI is provided by using port 80. During the deployment, the Server Load Balancer instance is created, and the blockchain

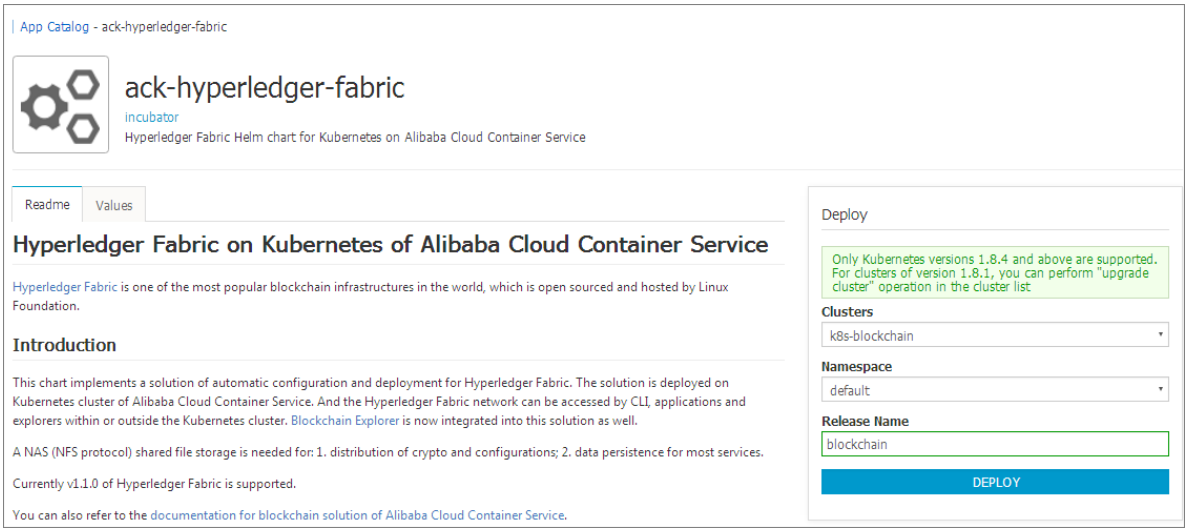
Parameter	Description
	explorer function based on Web UI is provided by using port 80. The default value is true.
logService.enabled	Whether or not to enable the support for Alibaba Cloud Log Service. The default value is false. For more information about Alibaba Cloud Log Service, see Collect Kubernetes cluster logs by using Alibaba Cloud Log Service. ########## *Kubernetes* ####
logService.region	Specify the region in which the Log Service project resides if logService.enabled is set to true. Set the parameter value as per your needs. Set the parameter value as per your needs. For more information about Alibaba Cloud Log Service, see Collect Kubernetes cluster logs by using Alibaba Cloud Log Service. ########## *Kubernetes* ####
logService.userID	Specify the user ID of the Alibaba Cloud primary account if logService.enabled is set to true. For more information about Alibaba Cloud Log Service, see Collect Kubernetes cluster logs by using Alibaba Cloud Log Service. ########## *Kubernetes* ####
logService.machineGroup	Specify the machine group of the Log Service project if logService.enabled is set to true. For more information about Alibaba Cloud Log Service, see Collect Kubernetes cluster logs by using Alibaba Cloud Log Service. ########## *# Kubernetes* ####

Deploy blockchain network in Container Service console

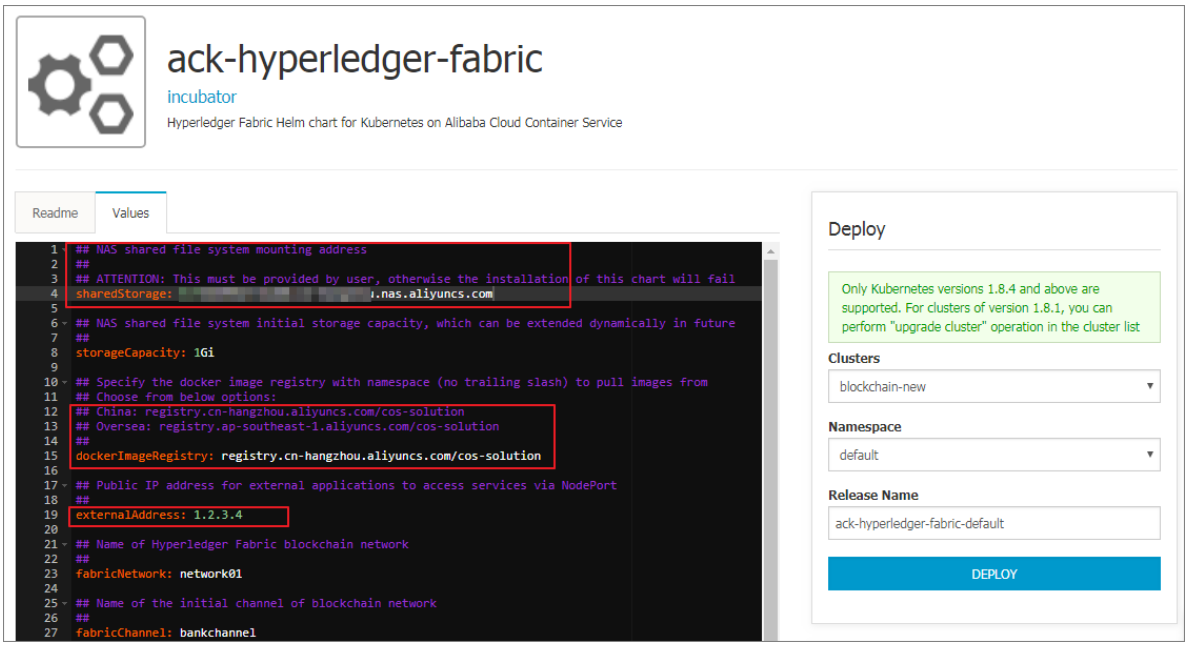
1. Log on to the Container Service console. Click Kubernetes > Store > **App Catalog** in the left-side navigation pane. Click **ack-hyperledger-fabric**.



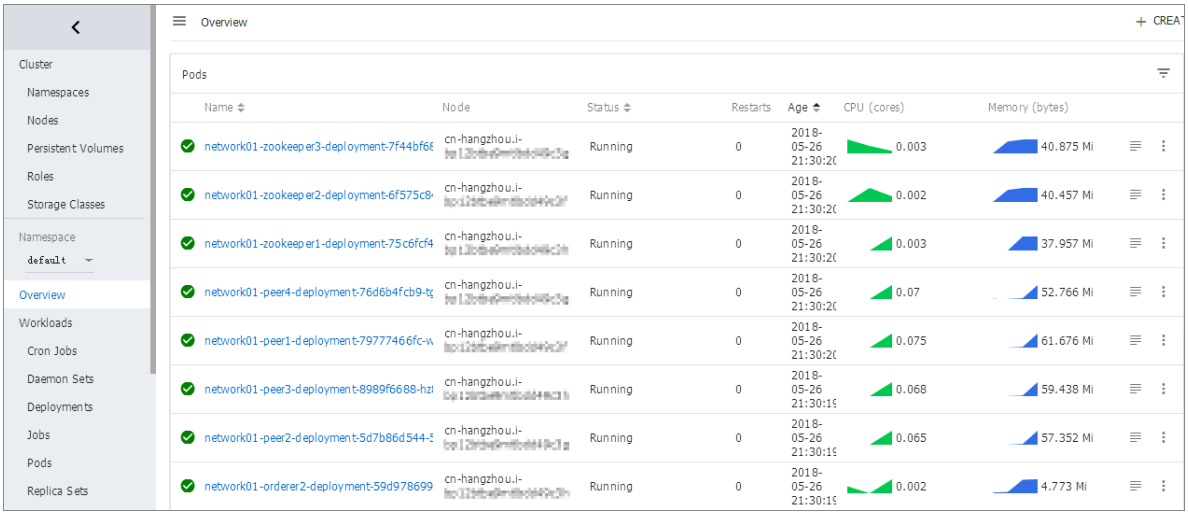
2. Select the cluster in which the blockchain network is to be deployed from the **Clusters** drop-down list and enter the **Release Name**.



3. Click the **Values** tab. View or modify the corresponding deployment parameters.



4. Click **Deploy**.
5. Enter the cluster dashboard. Check the status of the pods related to the blockchain network. Wait until the status of all the pods becomes Running.



Deploy blockchain network by using Helm commands

For more information about how to use Helm to deploy applications in Container Service Kubernetes clusters, see *## Helm ######*.

1. Log on to a master node of the Kubernetes cluster by using SSH. Use the root account and the password configured when creating the Kubernetes cluster.
2. Run Helm commands to deploy the blockchain network.

- To deploy the blockchain network by using the default parameter settings, run the following command:

```
helm install --name blockchain-network01 incubator/ack-hyperledger
-fabric
```

  Wherein, `--name` indicates the Helm release name of the blockchain network. You can set the parameter value as per your needs.

- To deploy the blockchain network to the specified namespace (such as network01), run the following command:

```
helm install --namespace network01 --name blockchain-network01
incubator/ack-hyperledger-fabric
```

  Wherein, `--namespace` indicates the namespace in which the blockchain network is to be deployed. You can set the parameter value as per your needs. namespace in which the blockchain network is to be deployed. You can set the parameter value as per your needs.

- To deploy the blockchain network with simple or a small number of configuration parameters, use the set parameter to pass in the configuration parameter values. For example:

```
helm install --name blockchain-network01 --set "fabricChannel=
mychannel" incubator/ack-hyperledger-fabric
```

- To deploy the blockchain network with complex or a large number of configuration parameters, use the YAML file to pass in the configuration parameter values. For example:

```
helm install --values network01.yaml --name blockchain-network01
incubator/ack-hyperledger-fabric
```

  Wherein, `--values` indicates the YAML file for customizing configuration parameters. You can set the parameter values as per your needs. See the following example: file for customizing configuration parameters. You can set the parameter values as per your needs. See the following example:

```
# sample values yaml
fabricNetwork: network01
fabricChannel: tradechannel
orgNum: 3
ordererNum: 4
ordererDomain: shop
peerDomain: shop
externalAddress: 11.22.33.44
caExternalPortList: ["31054", "31064", "31074"]
ordererExternalPortList: ["31050", "31060", "31070", "31080"]
peerExternalGrpcPortList: ["31051", "31061", "31071", "31081", "
31091", "31101"]
```

```
peerExternalEventPortList: ["31053", "31063", "31073", "31083", "
31093", "31103"]
```

Check if the Run the following command to make sure the release status of the blockchain network is Deployed. Helm release of the blockchain network is successfully deployed.

Run the following command to make sure the release status of the blockchain network is Deployed.

3. Run the following command to check if all the node pods of the blockchain network are successfully running and make sure the status of all the pods in the blockchain network is Running.

```
kubectl get pod
```

- If a namespace is specified, for example, network01, run the following command:

```
kubectl get pod -n network01
```

- To listen to the pod status changes in watcher mode, run the following command:

```
kubectl get pod -w
```

4. Run the following command to check the blockchain network deployment status. If the status is 如果显示状态为 `DEPLOYED`, the blockchain network is successfully deployed.

```
helm list
```

Then, the blockchain network configuration and deployment are complete.

Naming rules for node services of blockchain network

For standard node types of Hyperledger Fabric, the service naming rule is as follows:

```
<blockchain network name>--<node type><serial number>
```

For example:

```
network01-peer1
network01-peer2
network01-orderer1
network01-ca1
```

Although blockchain networks can be differentiated by namespaces in a Kubernetes cluster, the preceding service naming rule still uses a blockchain network name as the prefix to keep consistent with the blockchain solution of swarm clusters.

A service name must be consistent with the node name in the blockchain certificate and key. This type of service name (together with the service port) can be used to directly access blockchain

applications or CLI deployed in the same Kubernetes cluster. The external address is not required .

Problem diagnosis

This section introduces some common ideas, methods, and tools for diagnosing problems and errors that may occur during blockchain configuration, deployment, and access.

First, check if #### all preparations described in Environment preparations have been completed correctly. For example, whether or not the shared file system is correctly mounted to each worker node and NFS 4.0 protocol is specified.

Then, use Kubernetes-related commands, such as **`kubectl describe pod`**, **`kubectl logs`**, and **`kubectl get pod -o yaml`**, to view deployment events and output logs.

In addition, to assist in fault diagnosis and problem troubleshooting, the blockchain solution deploys a custom fabric-utils container in the blockchain network. This container integrates with common basic tools, such as telnet, ping, nslookup, and curl. telnet, ping, nslookup, and curl. You can run the following kubectl command on the master node to enter the fabric-utils container and use suitable tools to diagnose and analyze the problems. container and use suitable tools to diagnose and analyze the problems. For example:

```
kubectl exec -it fabric-utils-pod bash
```

Finally, for problems or errors related to Hyperledger Fabric, you can search Hyperledger Fabric official documents, StackOverFlow, or Google/Bing/Baidu for relevant information or solutions. StackOverFlow, or Google/Bing/Baidu for relevant information or solutions.

7 View blockchain network logs

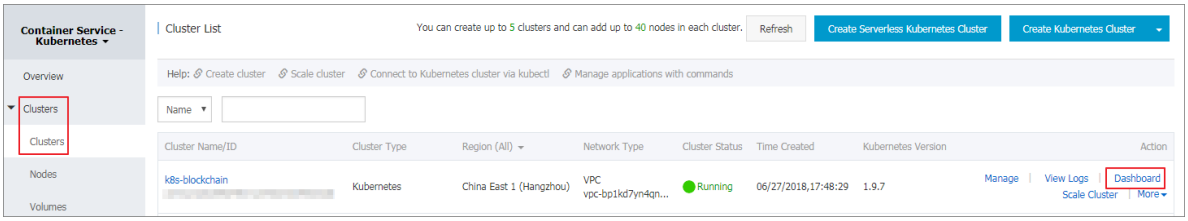
Logs generated when the blockchain network is running are output as container logs of Peer, Orderer, CA, Kafka, and ZooKeeper node types. The Container Service blockchain solution allows you to view these logs in the Container Service console, by using Kubernetes commands, or by means of Alibaba Cloud Log Service. This document introduces these three methods in details.

View logs in Container Service console

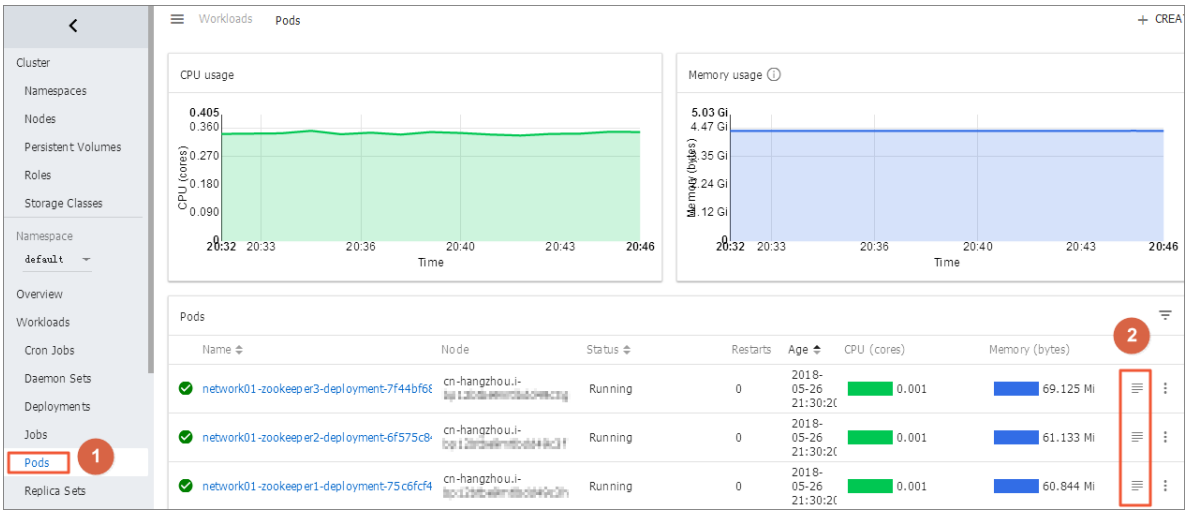
The Container Service console allows you to conveniently view logs on GUI. After deploying the blockchain network, you can follow these steps to view the container logs of the corresponding node.

Procedure

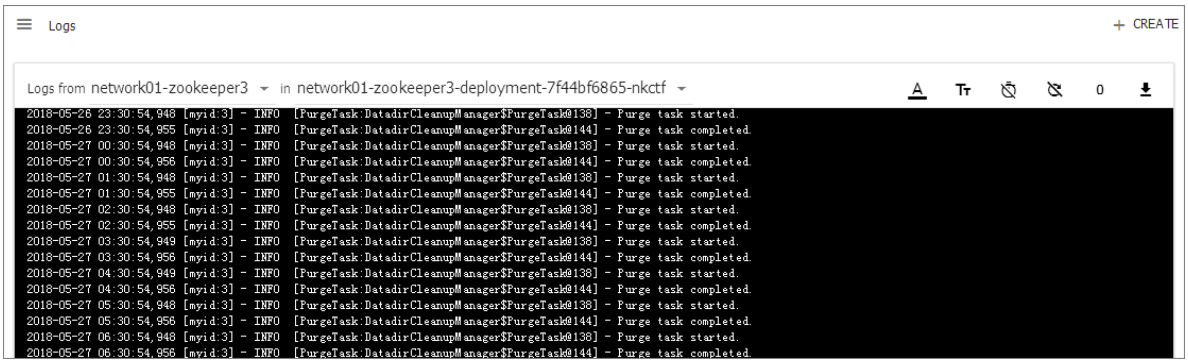
1. Log on to the *Container Service console*.
2. Click **Kubernetess** > **Clusters** in the left-side navigation pane. Click Dashboard at the right of the cluster.



3. Click **Pods** in the left-side navigation pane and then click the **Logs** icon.



4. View the log details.



5. You can also click the name of a pod and click **LOGS**. Then, view the log details.

View logs by using Kubernetes commands

You can also use the standard **`kubectl logs`** command to view the Container Service logs of the blockchain network.

Procedure

1. Log on to the Container Service console. Click Kubernetes > Clusters in the left-side navigation pane. Click **Manage** at the right of the cluster in which the blockchain network is deployed. Obtain the **Master node SSH IP address**.
2. Log on to the master node of the Kubernetes cluster by using SSH. Enter the username `root` and the password configured when creating the cluster.
3. Run the **`kubectl get pod`** command to obtain the pod list and select the name of the pod whose logs you want to view.
4. Run the **`kubectl logs pod name`** command to view the logs.
5. If a pod contains multiple containers, you can run the **`kubectl logs pod name container name`** command to view the logs of a container.

View logs by using Alibaba Cloud Log Service

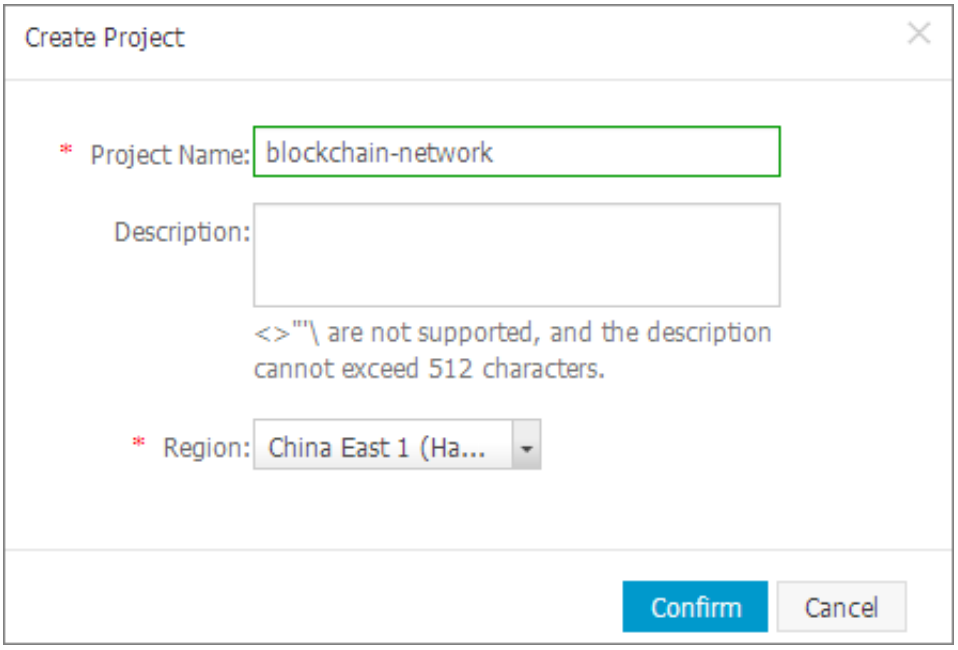
The log function of the Container Service console and the Kubernetes commands can basically meet the common requirements of log viewing. However, enterprise level requirements may include advanced functions such as log storage, real-time query and analysis, alarm, and visualized report. Then, you can integrate with *Alibaba Cloud Log Service* for expansion.

Container Service blockchain solution supports integrating with Alibaba Cloud Log Service. The basic procedures are as follows. For more information about integrating Container Service Kubernetes clusters with Alibaba Cloud Log Service, see *Log management*.

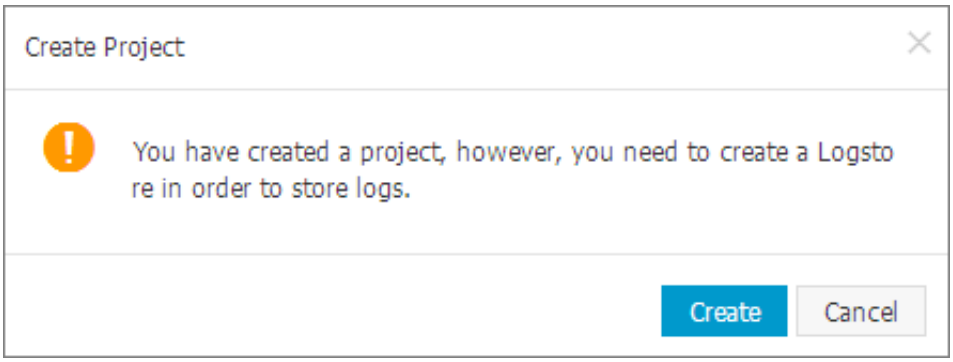
You may be charged by using Alibaba Cloud Log Service. For more information, see *Log Service billing* method.

Procedure

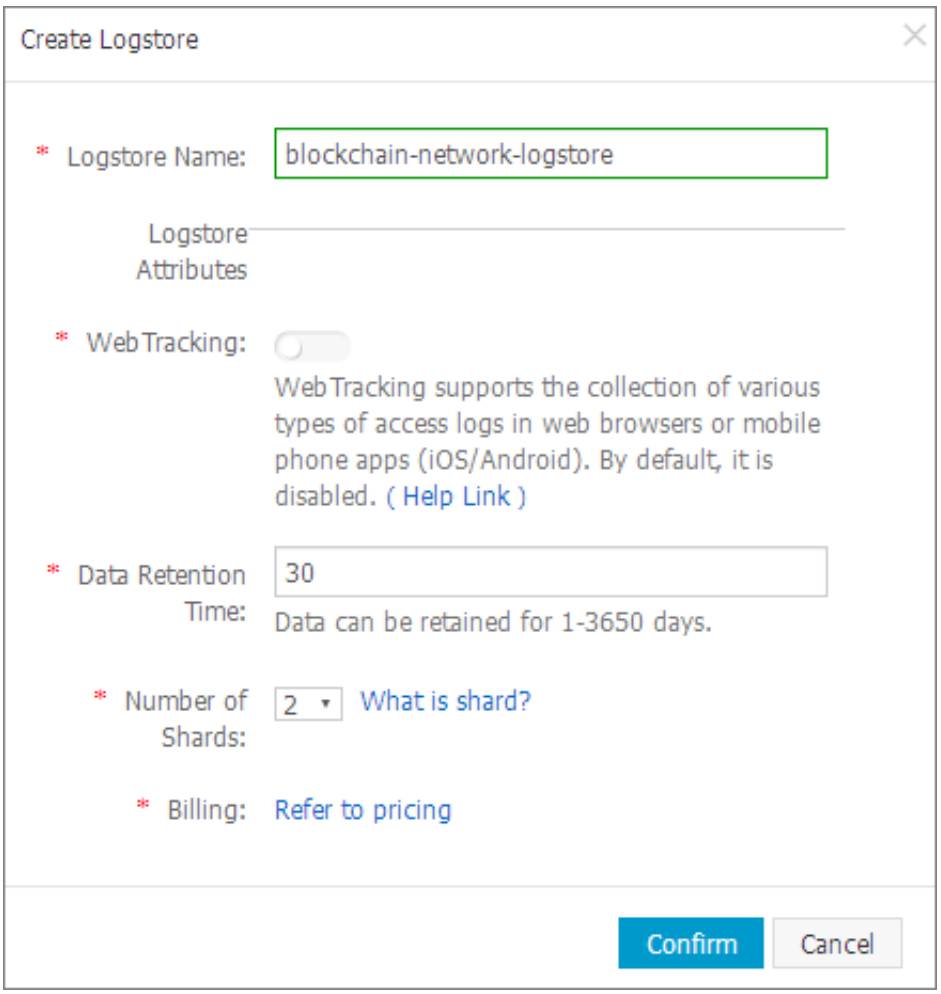
1. Log on to the *Log Service console* and activate Log Service as instructed.
2. Click **Create Project** in the upper-right corner.
3. Enter the Project Name, and select the Region in which the blockchain network resides. Then, click **Confirm**.



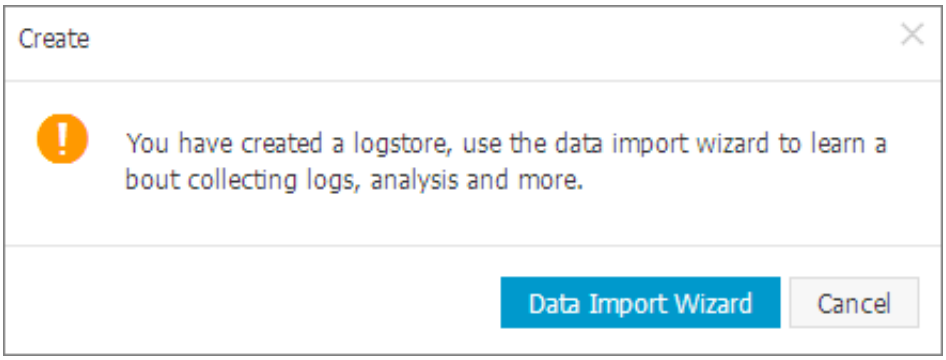
4. Click **Create** in the displayed dialog box to create the Logstore.



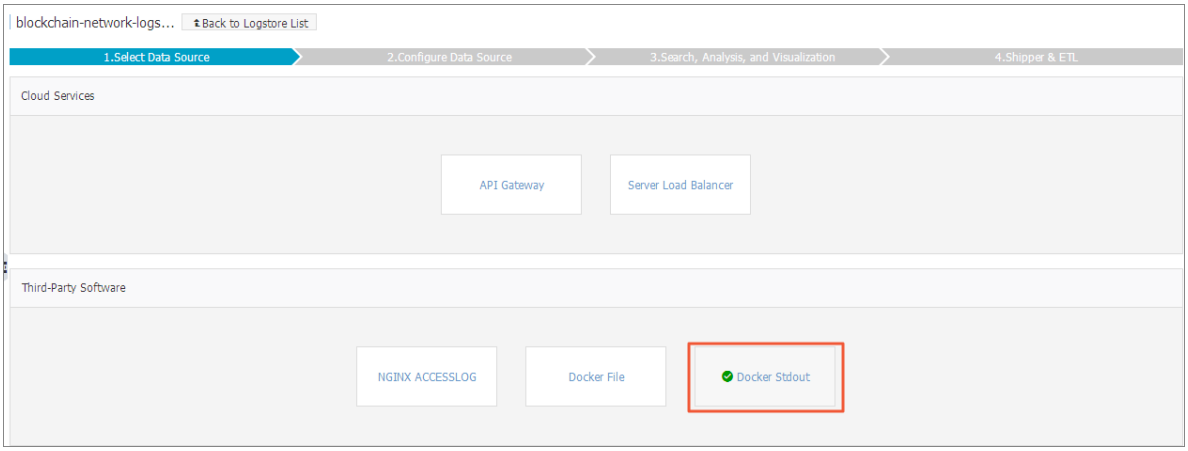
5. The Create Logstore dialog box appears. Enter the Logstore name. Complete the other configurations as per your needs. Then, click **Confirm**.



6. Click **Data Import Wizard** in the displayed dialog box.



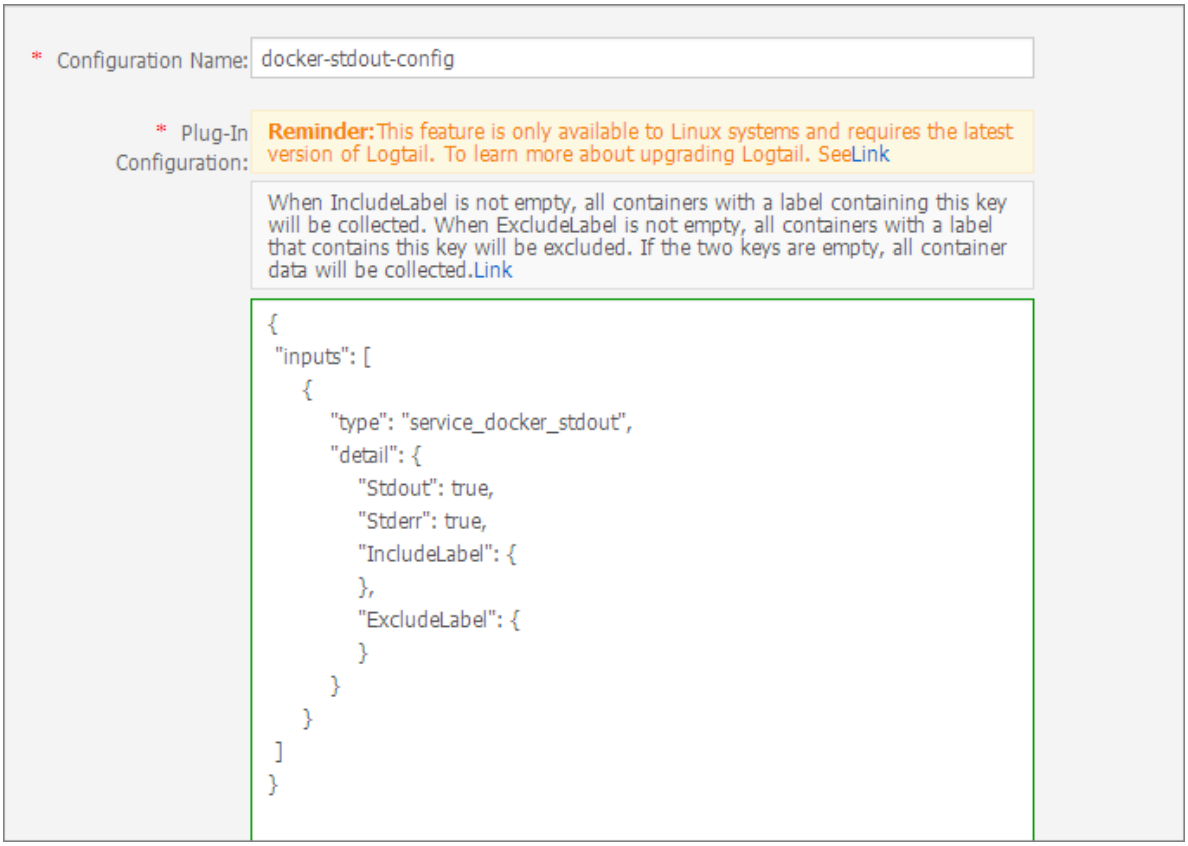
7. Select Docker Stdout under Third-Party Software. Then, click **Next**.



8. In the Plug-In Configuration field, enter the following example configuration. For more information about the configurations, see *Container standard output*. Then, click **Next**.

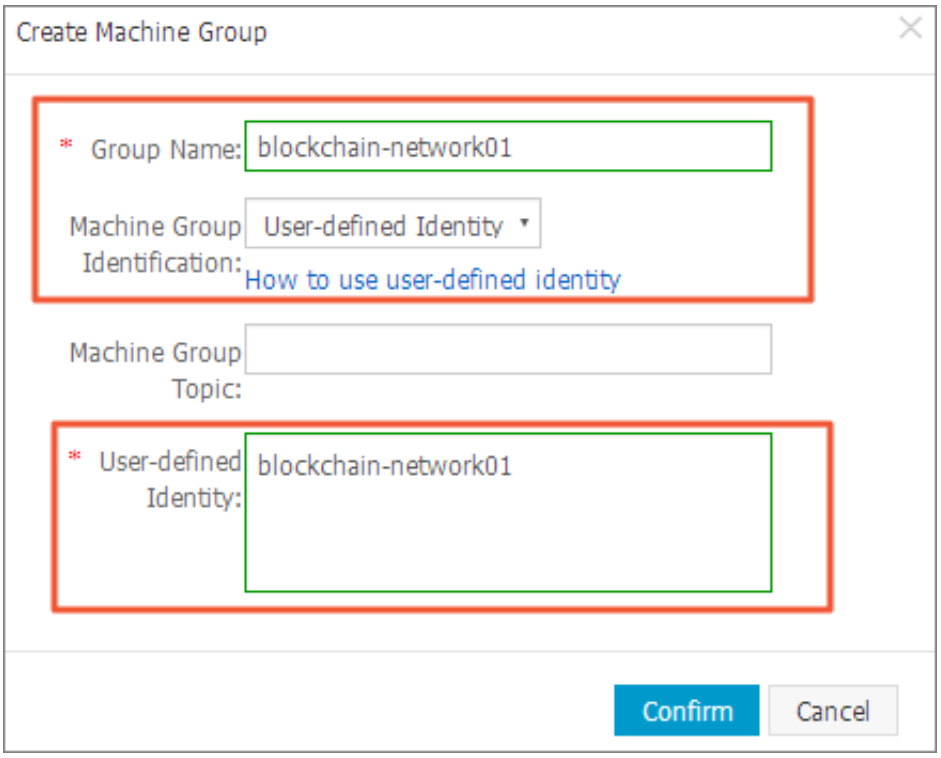
```
{
  "inputs": [
      {
          "type": "service_docker_stdout",
          "detail": {
              "Stdout": true,
              "Stderr": true,
              "IncludeLabel": {
              },
              "ExcludeLabel": {
              }
          }
      }
  ]
```

```
}
```

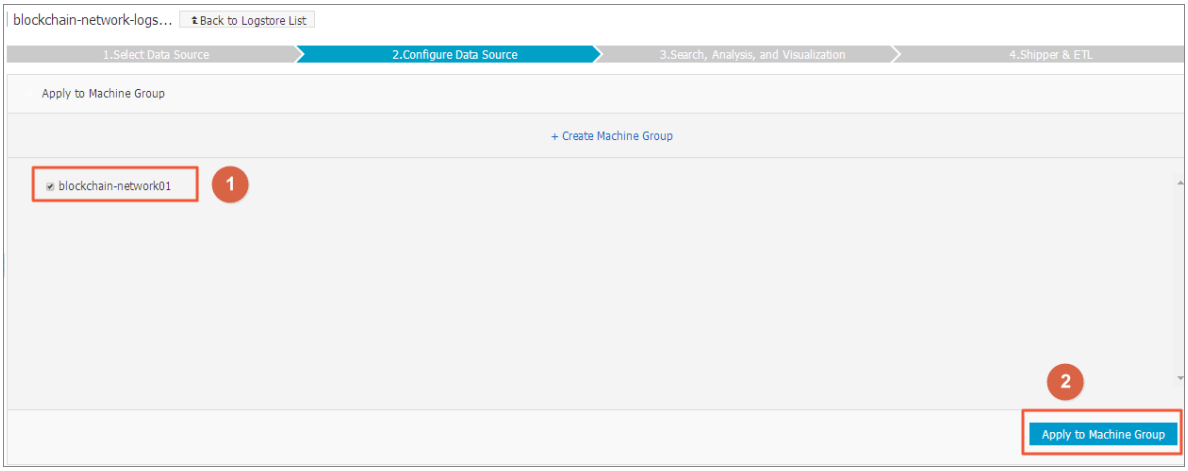


9. Click **Create Machine Group**。

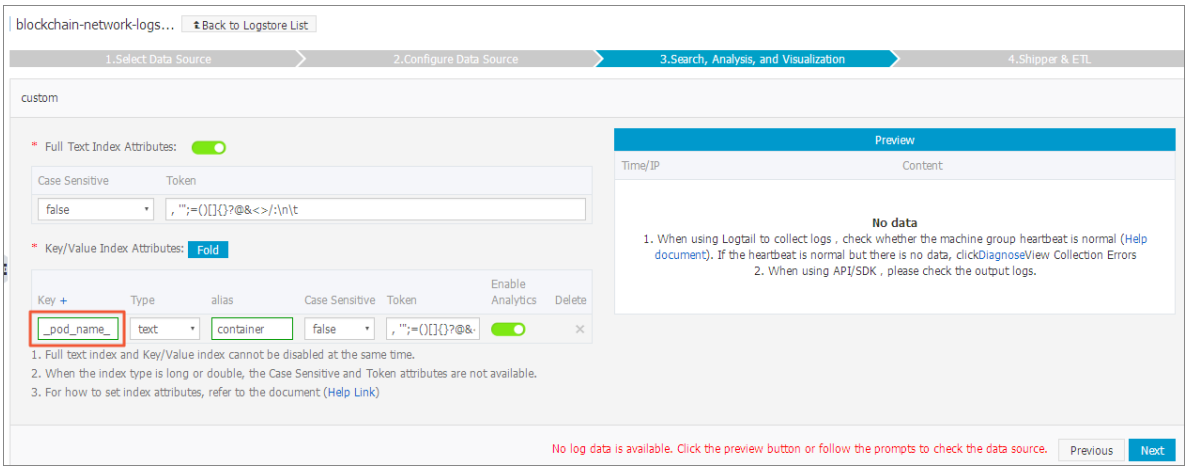
   The Create Machine Group dialog box appears.Enter the custom machine group name in the Group Name field. Select **User-defined Identity** from the Machine Group Identification drop-down list. In the User-defined Identity field, enter the same content as that in the Group Name field. Then, click **Confirm** .



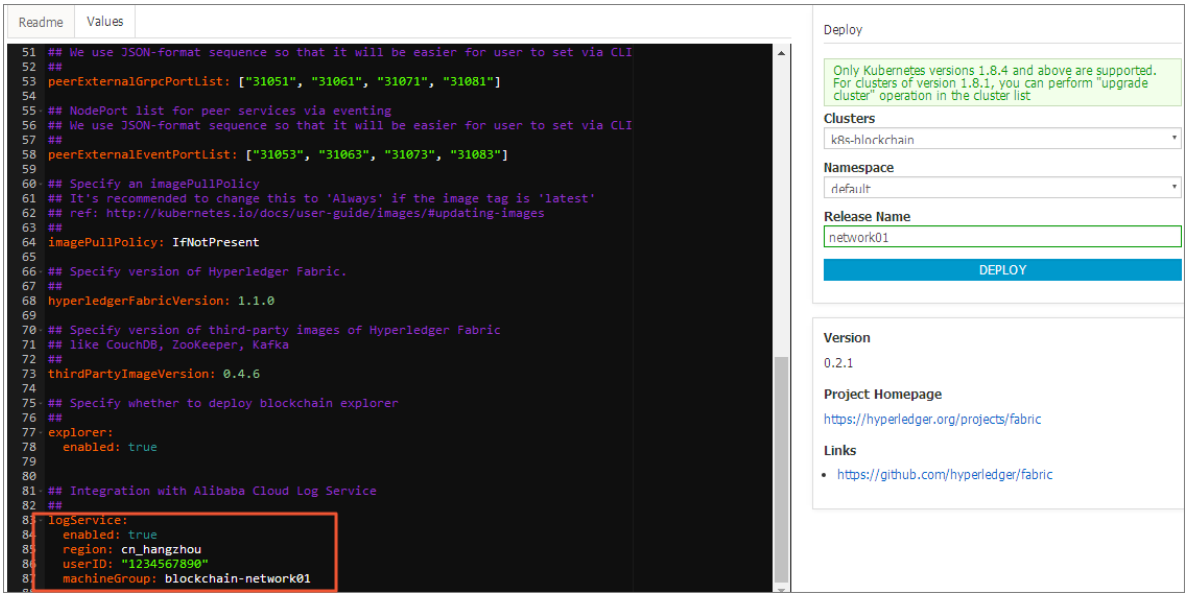
10. Select the created machine group and then click **Apply to Machine Group**.



11. Add the key (for example, _pod_name_) for creating the index as per your needs. After completing the configurations, click **Next**.



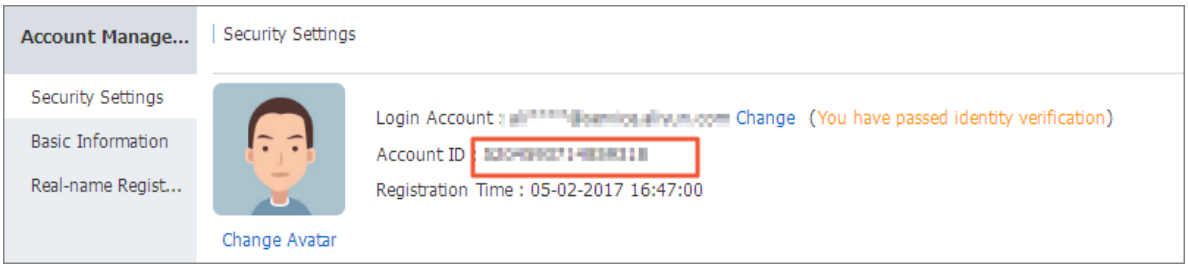
12. Click **Confirm**. Then, the creation and initial configuration of Alibaba Cloud Log Service are complete. Deploy a new blockchain network by using the blockchain solution
13. and configure the parameters integrated with Log Service on the Values page of the blockchain solution.



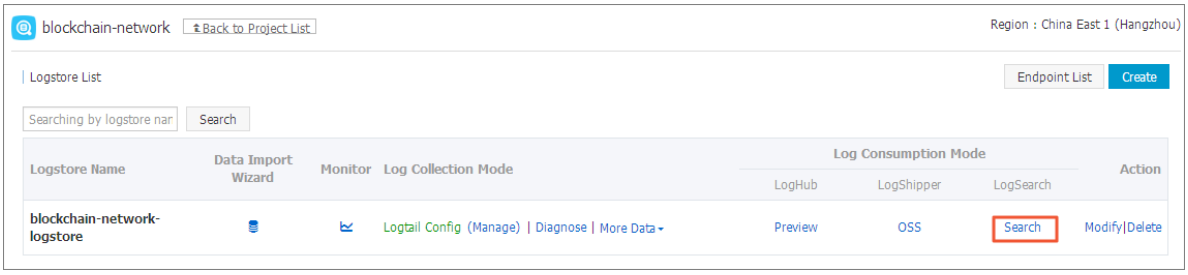
Set the parameter **`enabled`** to true, which indicates to enable Log Service. Set the parameter machineGroup to the user-defined identity configured in the machine group. In this example, it is **`machineGroup`** to the user-defined identity configured in the machine group. In this example, it is blockchain-network01.

To set the parameter region, see *Linux* to search for the corresponding installation commands and the region ID. For example, `cn_hangzhou` indicates to write logs from the Alibaba Cloud intranet in the region Hangzhou to Log Service and no Internet bandwidth is consumed.

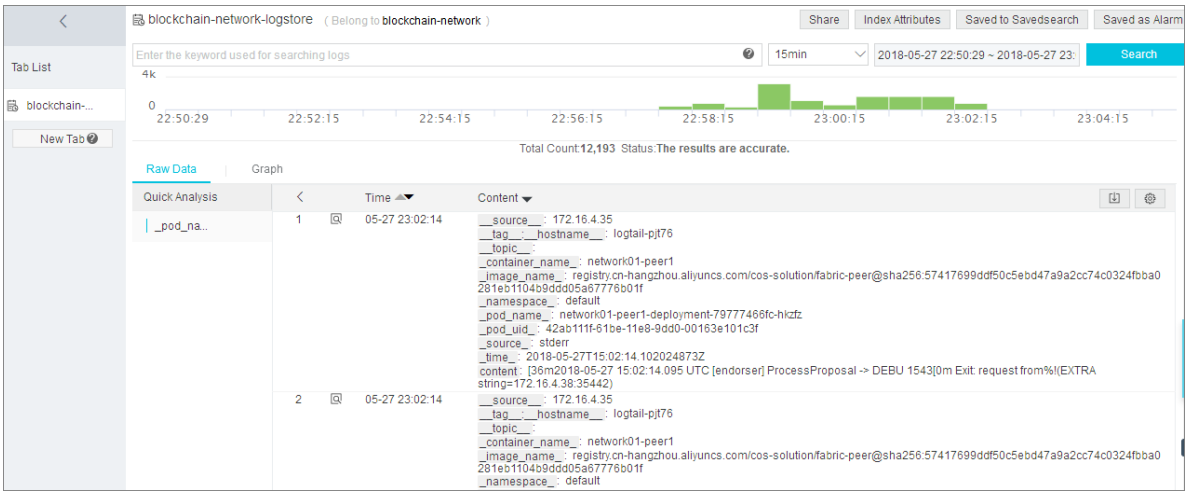
To set the parameter userID, see the following figure for reference.



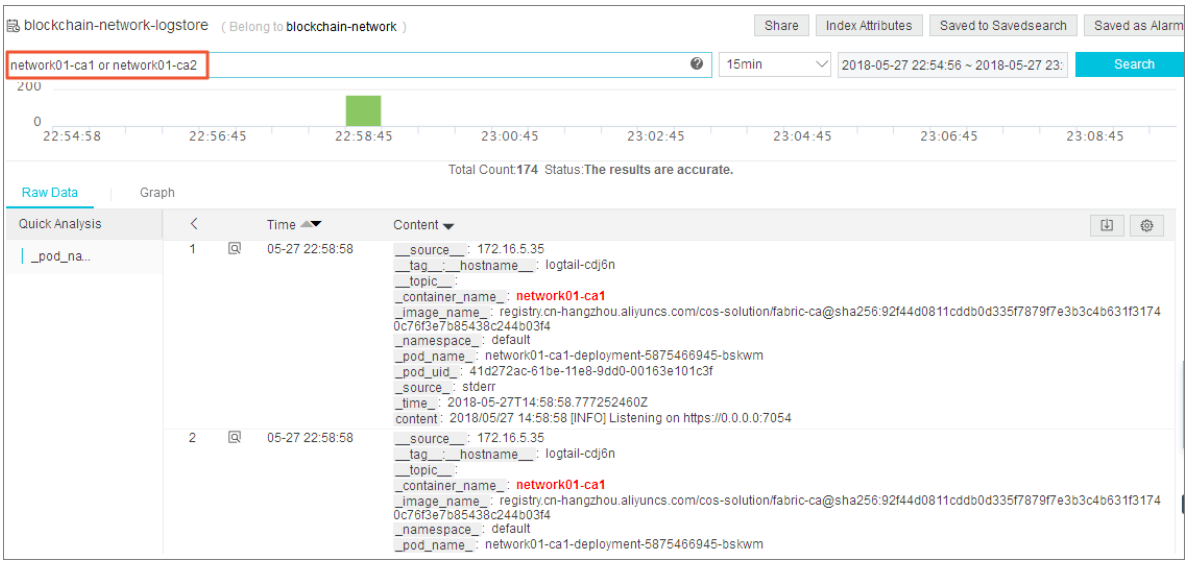
14.Log on to the Log Service console. Click the project name and then click **Search** at the right of the



Logstore.



15.Log Service supports complex queries. For more information about the query syntax and other advanced functions of Log Service, see *Query syntax*.


blockchain-network-logstore (Belong to blockchain-network)
Share
Index Attributes
Saved to Savedsearch
Saved as Alarm
network01-ca1 or network01-ca2
15min
2018-05-27 22:54:56 ~ 2018-05-27 23:
Search
200
0
22:54:58
22:56:45
22:58:45
23:00:45
23:02:45
23:04:45
23:06:45
23:08:45
Total Count:174 Status:The results are accurate.
Raw Data
Graph
Quick Analysis
_pod_na...
Time
Content
1
05-27 22:58:58
__source__: 172.16.5.35
__tag__:__hostname__: logtail-cdj6n
__topic__:
_container_name_: network01-ca1
_image_name_: registry.cn-hangzhou.aliyuncs.com/cos-solution/fabric-ca@sha256:92f44d0811cddb0d335f7879f7e3b3c4b631f31740c76f3e7b85438c244b03f4
namespace: default
_pod_name_: network01-ca1-deployment-5875466945-bskwm
_pod_uid_: 41d272ac-61be-11e8-9dd0-00163e101c3f
source: stderr
time: 2018-05-27T14:58:58.777252460Z
content: 2018/05/27 14:58:58 [INFO] Listening on https://0.0.0.0:7054
2
05-27 22:58:58
__source__: 172.16.5.35
__tag__:__hostname__: logtail-cdj6n
__topic__:
_container_name_: network01-ca1
_image_name_: registry.cn-hangzhou.aliyuncs.com/cos-solution/fabric-ca@sha256:92f44d0811cddb0d335f7879f7e3b3c4b631f31740c76f3e7b85438c244b03f4
namespace: default
_pod_name_: network01-ca1-deployment-5875466945-bskwm

8 Access blockchain network

After configuring and deploying the blockchain network on Alibaba Cloud, the blockchain developer or administrator can access the blockchain network and use different blockchain node services. Common access methods are as follows:

- The developer or administrator remotely connects to the blockchain node container, and tests or manages the blockchain network by using CLI.
- The blockchain applications connect to the blockchain network services such as CA, Orderer, and Peer for transactions and service calls based on the blockchain.
- The blockchain administration and monitoring tools connect to the blockchain network, managing and monitoring the blockchain network and each node in the graphical or automated way.

These blockchain access methods, applications, and codes can be developed and deployed according to your own business and technical requirements. They can be either deployed as container applications in the container cluster, together with the blockchain network, or deployed in your own environment that you can access the blockchain network from outside the container cluster. For the second method, create an Internet IP address or a Server Load Balancer instance for the blockchain network in advance and configure security group rules for the external port NodePort to allow external access.

The following documents provide simple examples and instructions for main blockchain access methods to help you better understand and develop related applications and tools. At the same time, the blockchain solution provides configuration files that can be downloaded with one click (including certificates, keys, and blockchain network configuration files required to connect to blockchain services) to accelerate the development and testing processes of blockchain applicatio ns, administration, and monitoring tools.

- *####IP#####*
- *#CLI#######*
- *############*
- *##############*

9 Access blockchain network by using CLI

Context

The blockchain solution creates CLI containers during the blockchain network deployment. CLI containers connect to the Peer and Orderer nodes of the blockchain network in command line mode, run CLI commands supported by Hyperledger Fabric, and satisfy the requirements for testing and managing the blockchain network.

To log on to a CLI container, run the command **`kubectl exec -it fabric-cli -n <namespace name> bash`** in an environment that supports kubectl commands. The CLI container provides a standard end-to-end CLI test script based on Hyperledger Fabric. To modify the test script, find the file in the path */data/fabric/<blockchain network name>/cli/cli-test.sh* on any Elastic Compute Service (ECS) instance.

Procedure

1. Use SSH to log on to a master node of the Kubernetes cluster as the root account. For how to obtain the address, see *Environment preparations*. You can also access an environment that supports managing Kubernetes clusters remotely by using kubectl.
2. Run the command **`kubectl exec -it -n <namespace name> fabric-cli bash`** to enter the CLI container, for example, **`kubectl exec -it -n network01 fabric-cli bash`**.
3. Run the test script *./cli-test.sh* to start the CLI test.
4. After each step is complete, the test pauses so that you can view the execution process and result. The test continues after you press any key. After the test script is executed successfully, information similar to the following one is displayed:

```
Query Result: 90
 2017-11-12 09:22:31.452 UTC [main] main -> INFO 007 Exiting.....
 ===================== Query on PEER4 on channel 'bankchannel' is
successful =====================
 Press any key to continue...
 ===================== All GOOD, End-2-End execution completed
 =====================
```

Note:

For the same blockchain network, CLI sample and application sample cannot run at the same time. Select to run only one type of sample for each blockchain network.

10 Clean up blockchain environment

After you complete the development and testing activities, clean up the blockchain environment if the deployed blockchain network is no longer in use or you must redeploy the blockchain network.

Use Helm to delete blockchain network

1. Use SSH to log on to a master node of the Kubernetes cluster as the root account.
2. Run the **`helm list`** command to view the Helm release name of the blockchain network.
3. Run the command **`helm delete --purge <Helm release name of the blockchain network>`** to delete the blockchain network. For example, **`helm delete --purge blockchain-network01`**.

Wait a few minutes (depending on the number of nodes in the blockchain network) until the helm delete command is complete and the result is returned. Then, the services and containers of all blockchain nodes are deleted from the Kubernetes cluster. The chaincdoe containers related to the blockchain network are also automatically deleted from all worker nodes.

Data directory of blockchain network

When the blockchain network is deleted, the data directory of the blockchain network in the shared file storage is automatically cleaned up for recreating the blockchain network. For security reasons, the data directory is cleaned up by adding the suffix -deleted-timestamp to the original directory name. For example, -deleted-2018-03-17-160332. In this way, you can reuse the data by deleting the suffix. To completely delete the data directory, manually use the rm command or use the automated script to release storage space with regular cleanup.

11 Access blockchain network by using applications

Prerequisites

- The blockchain network is configured and deployed in a Kubernetes cluster.
- The Internet IP address and external port are configured if the application is deployed outside container clusters.

Context

After creating the blockchain network by using the Alibaba Cloud Container Service blockchain solution, you can access services on the blockchain network by using blockchain applications based on Hyperledger Fabric SDKs. The blockchain solution supports the Connection Profile function from the 1.1 Hyperledger Fabric version.

- The blockchain application can be deployed on the Alibaba Cloud container cluster together with the blockchain network. In this mode, the application directly accesses services by using the name and port of each blockchain service.
- The blockchain application can also be deployed outside Alibaba Cloud container clusters. In this mode, the application accesses services by using the external address of the blockchain network and the external port of each service.

In this example, deploy the blockchain application outside Alibaba Cloud container clusters. The provided application is a balance transfer application developed and adapted based on Hyperledger Fabric.

You can use the application provided in this example, the official Hyperledger Fabric example application (for example, *fabric-samples*), or self-developed blockchain application. Adapt the application by referring to the source codes of the application provided in the example to access the Alibaba Cloud Container Service blockchain network.

Adapt the existing blockchain application as follows:

- Directly use the script download-from-fabric-network.sh that is provided in the sample code to automatically download blockchain network configurations from the newly deployed blockchain network with one click, including certificates, keys, and blockchain network configuration files (basically the config.json and network-config.yaml).

- Make sure that the blockchain application can be correctly loaded to the preceding blockchain network configurations.
- If the blockchain application directly uses the channel name, external address, node name, or domain name of the blockchain, replace them with parameters in the configuration files (config.json and network-config.yaml) to keep the configurations of the application consistent with those of the target blockchain network.

Procedure

1. Download the source codes of the blockchain application provided in the example to the local development environment. The command is as follows:

   ```
   git clone https://github.com/AliyunContainerService/solution-blockchain-demo.git
   ```

2. Perform subsequent operations based on the README document of the blockchain application provided in the example.

   Chinese version: *https://github.com/AliyunContainerService/solution-blockchain-demo/blob/master/balance-transfer-app/README.cn.md*

   English version: *https://github.com/AliyunContainerService/solution-blockchain-demo/blob/master/balance-transfer-app/README.md*

   Note:
   For the same blockchain network, CLI sample and application sample cannot run at the same time. Select to run only one type of sample for each blockchain network.

13 Blockchain network restart and data recovery

Container Service blockchain solution supports restarting the blockchain network. At the same time, the original blockchain configuration and data can be reused.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- Blockchain network is already running on the Kubernetes cluster. For more information, see *Quick start*.
- You have connected to the Kubernetes cluster by using SSH, see *Access Kubernetes clusters by using SSH*.

Procedure

1. Use SSH to log on to a master node of the Kubernetes cluster as the root account.
2. Run the **helm delete --no-hooks --purge <blockchain network Helm Release name>** command to delete blockchain network.

   Note:
   Use the **--no-hooks** parameter to avoid deleting the original data directory.

   ```
   helm delete --no-hooks --purge network01 #In this example, the Helm
   Release name is network01
   release "network01" deleted
   ```

3. Use the same blockchain network name as the fabricNetwork variable value to create a new blockchain network by using the Container Service console or the Helm command line. Therefore, you can reuse the original data directory. For more information, see *Configure and deploy blockchain network*. An example of Helm command is as follows.

   ```
   helm install --name network01 --set "sharedStorage=029bb489d2-ikw80
   .cn-hangzhou.nas.aliyuncs.com" incubator/ack-hyperledger-fabric #
   Replace with your NAS mount address
   ```

   To back up the blockchain data directory, use the following command example to mount the NAS file system to ECS. In this example, blockchain network name is network01.

   ```
   mkdir /data
   ```

```
mount -t nfs -o vers=4.0 987a6543bc-abc12.cn-hangzhou.nas.aliyuncs.
com:/ /data #Replace with your NAS mount address
```

Note:

Now, you must back up the /data/fabric/network01 data directory.

Using this method, you can perform blockchain network data backup, migration, and recovery.