Alibaba Cloud
Aliyun Container for Kubernetes

solution

Issue: 20190801

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.	 Notice: 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]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Blockchain solution

1.1 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 V1.4 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.

1.2 Version history and upgrade instructions

This document lists the version history and the corresponding functional changes of the Alibaba Cloud Container Service blockchain solution, and the instructions during the upgrade process. The version number is the chart version number of the blockchain solution ack-hyperledger-fabric in the application directory.

Version history

Version 0.2.3

- Hyperledger Fabric is upgraded to V1.4.0.
- Explorer is upgraded to V0.3.8.

Version 0.2.2

- Improves availability of NAS file system mount.

> Note:
> You must enter the NAS mount address each time you create a blockchain network. It is unnecessary to mount the NAS to ECS in the environment preparation.

- Improves availability of blockchain node deployment.
- Blockchain log level parameterized configuration.
- Improves data cleanup, explorer startup, chaincode execution timeout.
- Unified naming convention for partial pods.

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.
  - More 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 the 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 hyperledge r
```

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

```
helm repo update
```

1.3 Quick start

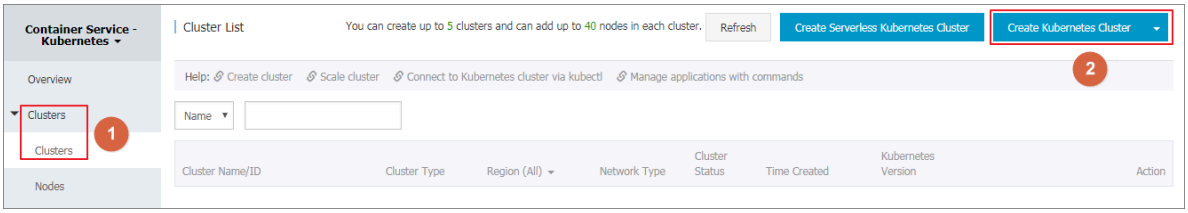
This document is a quick start guide for deploying the blockchain network from the beginning, in which most settings use the default or example values. For more information about the configurations, see the subsequent sections.

Limits

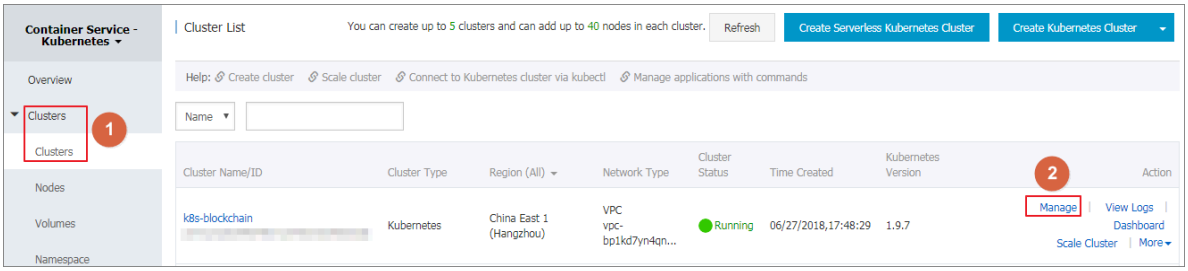
- You must register an account and activate Container Service.
- The container cluster and the NAS file system must be in the same region. Make sure the selected region supports both the Kubernetes cluster and the NAS file system (intersection of both). For the list of regions that support the NAS file system, see Products > Storage & CDN > NAS > > File System List.
- The usage of a file system only applies to the development and test phases of blockchain applications and solutions. For deployment in the production environment, contact us to further discuss specific business and technical needs and determine the most suitable method.

Step 1. Create a Kubernetes cluster

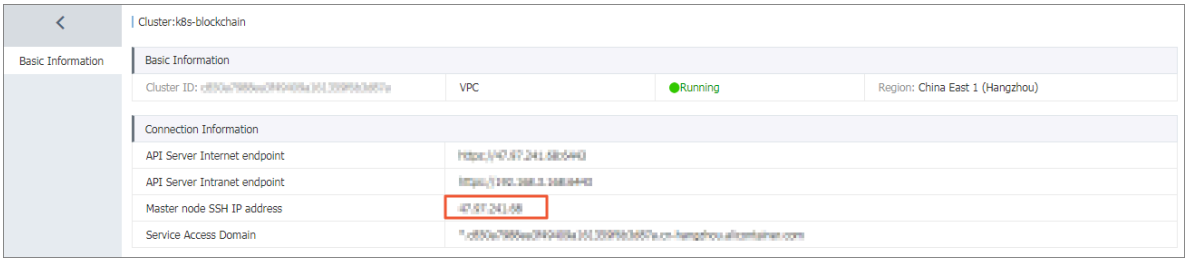
1. Log on to the Container Service console.
2. Under Kubernetes, click Clusters in the left-side navigation pane. Click Create Kubernetes Cluster in the upper-right corner.



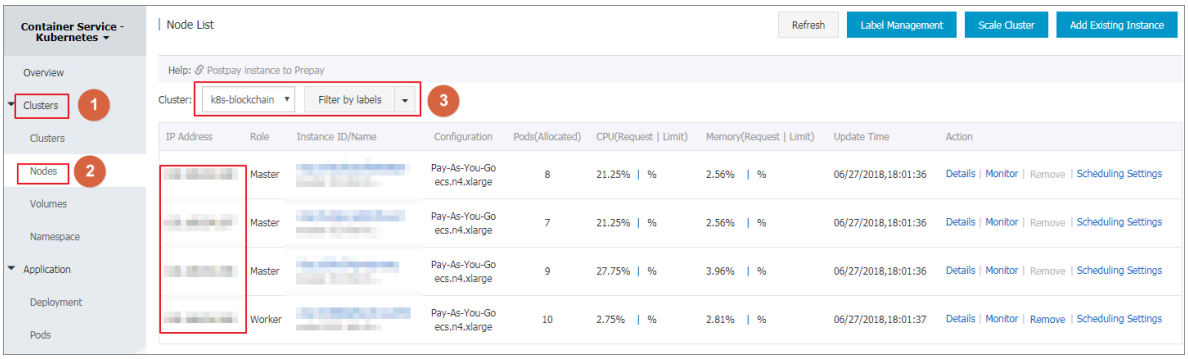
3. Configure the basic information for the cluster. In this example, complete the configurations as follows:
   - Enter the Cluster Name. For example, k8s-blockchain.
   - Select China East 1 (Hangzhou) as the Region.
   - Select China East 1 Zone B as the Zone.
   - Select Auto Create to create a Virtual Private Cloud (VPC) together with the Kubernetes cluster, or Use Existing to use an existing VPC.
   - Configure the Logon Password and Confirm Password.
   - Configure instance specifications and quantity. The blockchain network deployment is resource-intensive. Therefore, we recommend that you use the default configurations.
   - Select the Enable SSH access for Internet check box.
   - Click Create Cluster. Wait a few minutes until the cluster is successfully created.
4. On the Cluster List page, click Manage at the right of the created cluster (k8s-blockchain).



5. On the Basic Information page of the cluster, record the Master node SSH IP address, which is an Internet address.



6. On the Node List page, record the IP addresses (intranet addresses) of the cluster nodes (both master nodes and worker nodes).



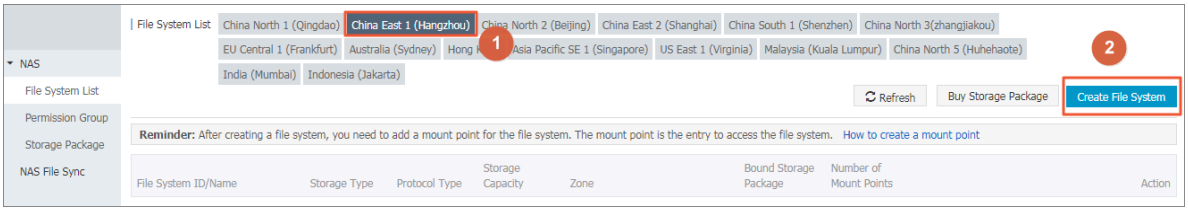
Step 2. Bind an EIP to a worker node

1. Log on to the Elastic IP Address (EIP) console.
2. Click Elastic IP Addresses in the left-side navigation pane. Click Create EIP.
3. Select the Region in which the cluster k8s-blockchain resides. Complete the other configurations as per your needs. Click Buy Now.
4. After activating the EIP, wait until the purchased EIP is displayed in the list and then click Bind at the right of the EIP.

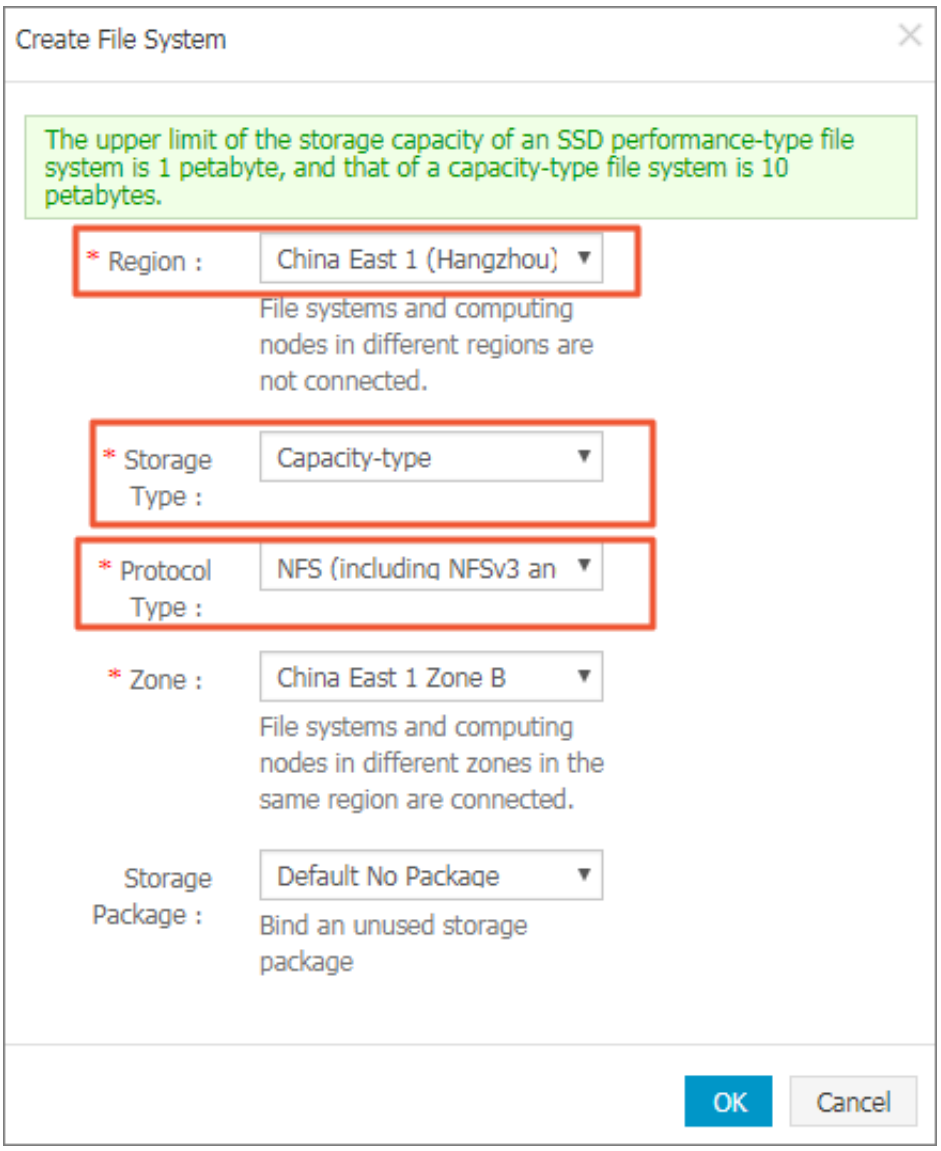
5. Select ECS Instance from the Instance Type drop-down list and select a worker node (the instance name begins with node). Then, click OK.
6. After binding the EIP to the worker node, record the IP address of the purchased EIP on the Elastic IP Addresses page.

Step 3. Create a file system and add a mount point

1. Log on to the NAS console.
2. Select China East 1 (Hangzhou) in the region list and click Create File System.

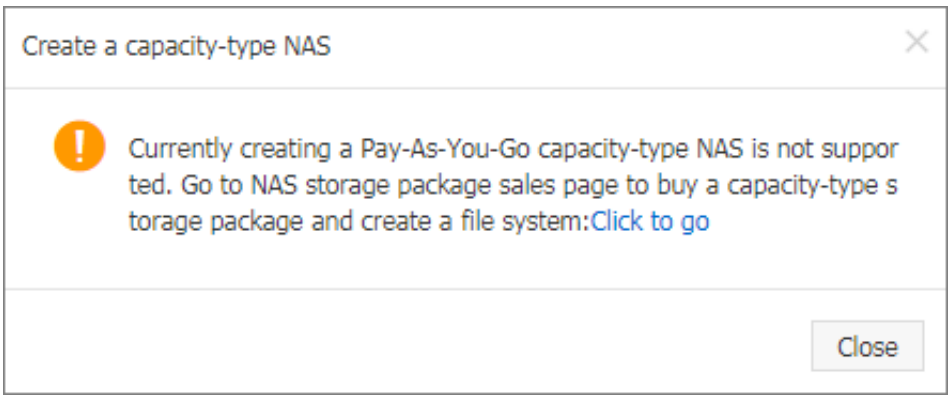


3. Complete the configurations in the displayed Create File System dialog box.

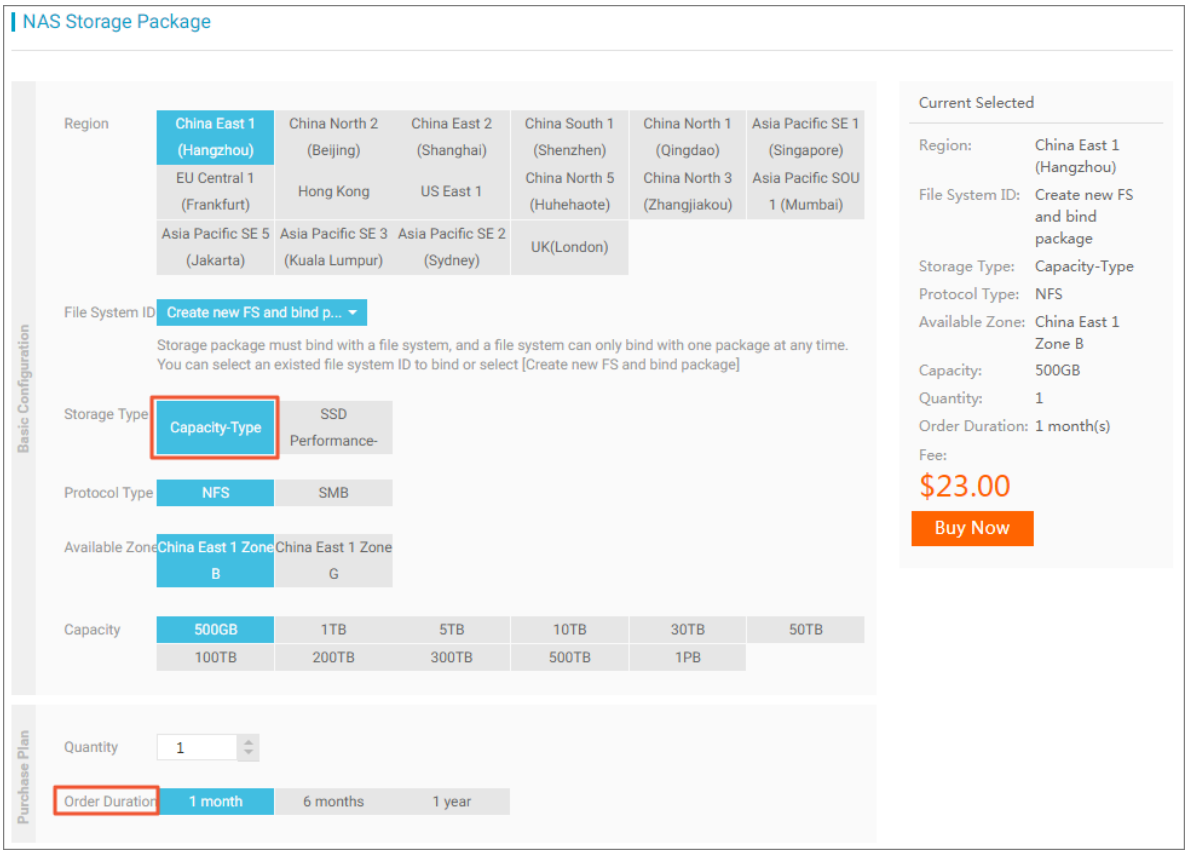


- Region: Select China East 1 (Hangzhou). Select the region in which the container cluster resides.
- Storage Type: In this example, select Capacity-type.
- Protocol Type: Select NFS.
- Zone: Select China East 1 Zone B. Different zones in the same region are interconnected.
- Click OK.

4. Click Click to go in the displayed dialog box.



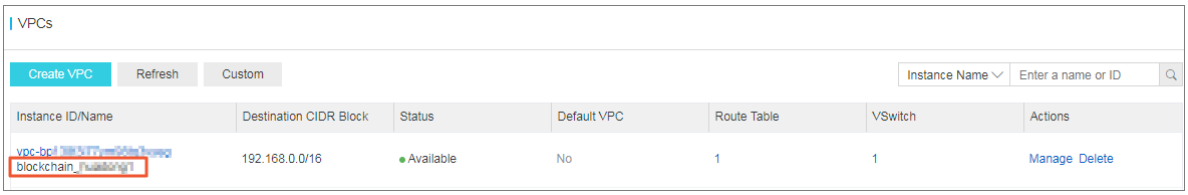
5. Configure the NAS storage package.



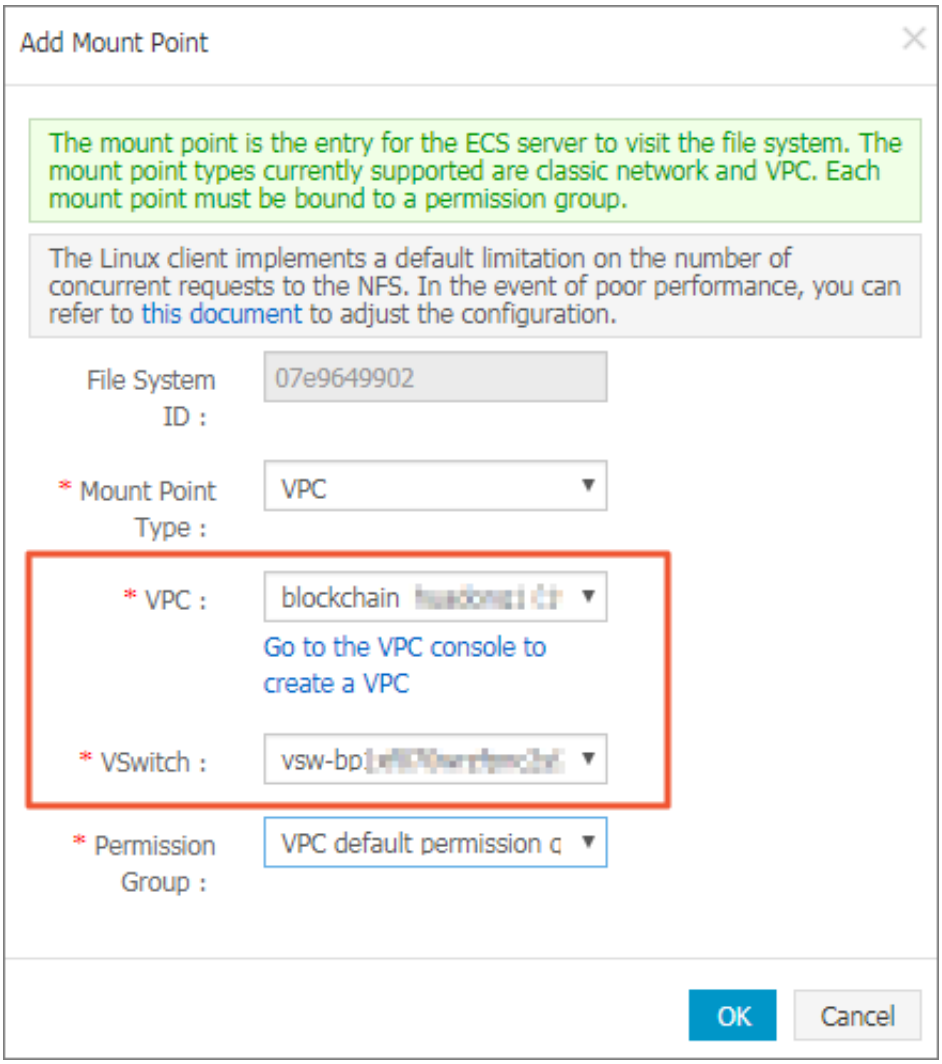
- Storage Type: Select Capacity-Type.
- Order Duration: In this example, select one month. Select the duration as per your needs.
- Click Buy Now.

6. If Use Existing is selected in the VPC field when you create the Kubernetes cluster, skip this step. If Auto Create is selected in the VPC field when you create the

Kubernetes cluster, go to the VPC console. Change the VPC instance name to one that is easy to identify, for example, blockchain_huadong1.

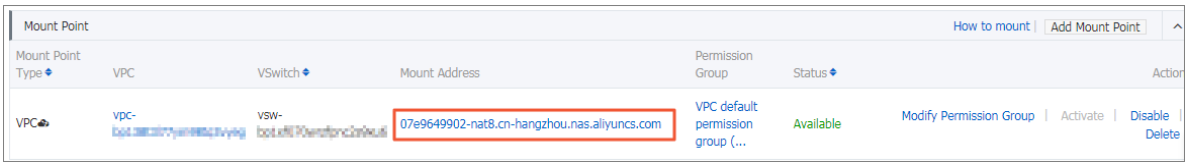


7. Click Add Mount Point at the right of the created file system. Complete the configurations in the displayed Add Mount Point dialog box.



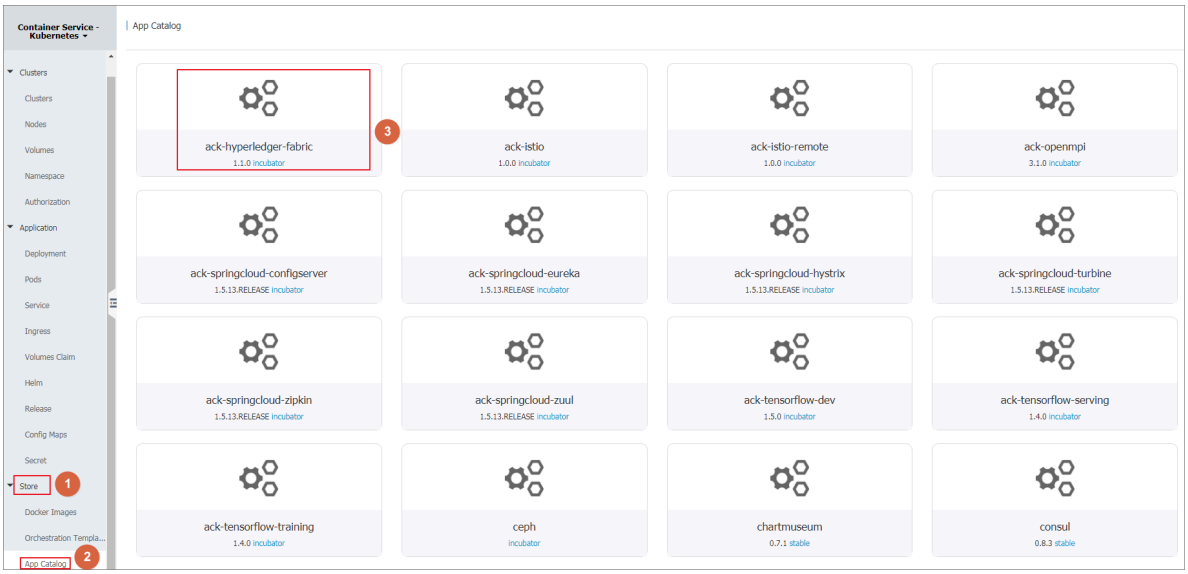
- Mount Point Type: Select VPC.
- VPC: Select the VPC used when creating the container cluster.
- VSwitch: Select the VSwitch used when creating the container cluster.
- Permission Group: Select VPC default permission group (allow all).
- Click OK.

8. On the File System List page, click Manage at the right of the file system. Record the Mount Address on the File System Details page.

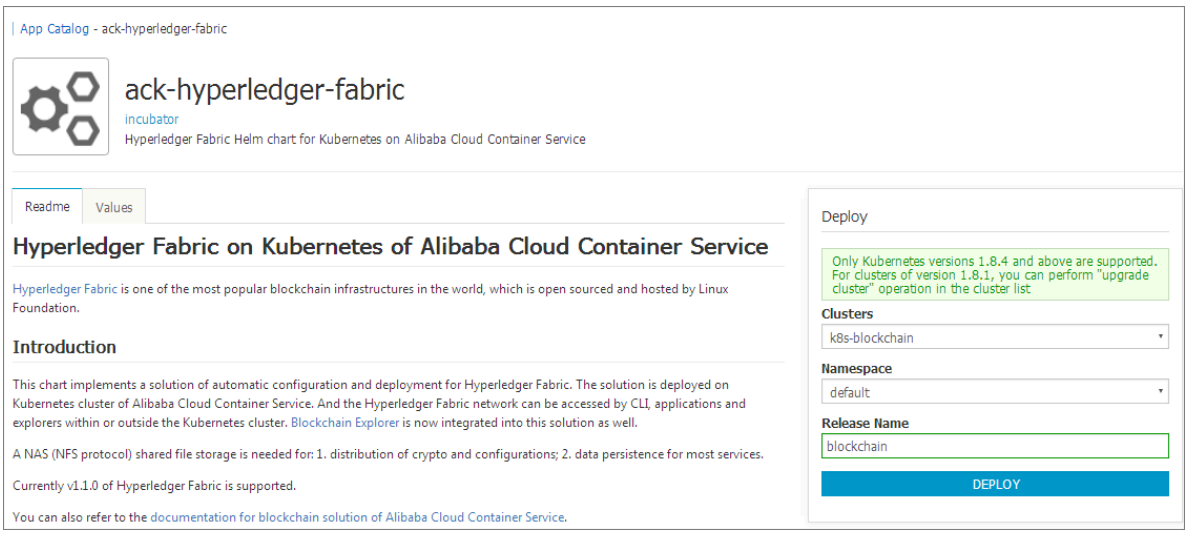


Step 4. Configure and deploy blockchain network

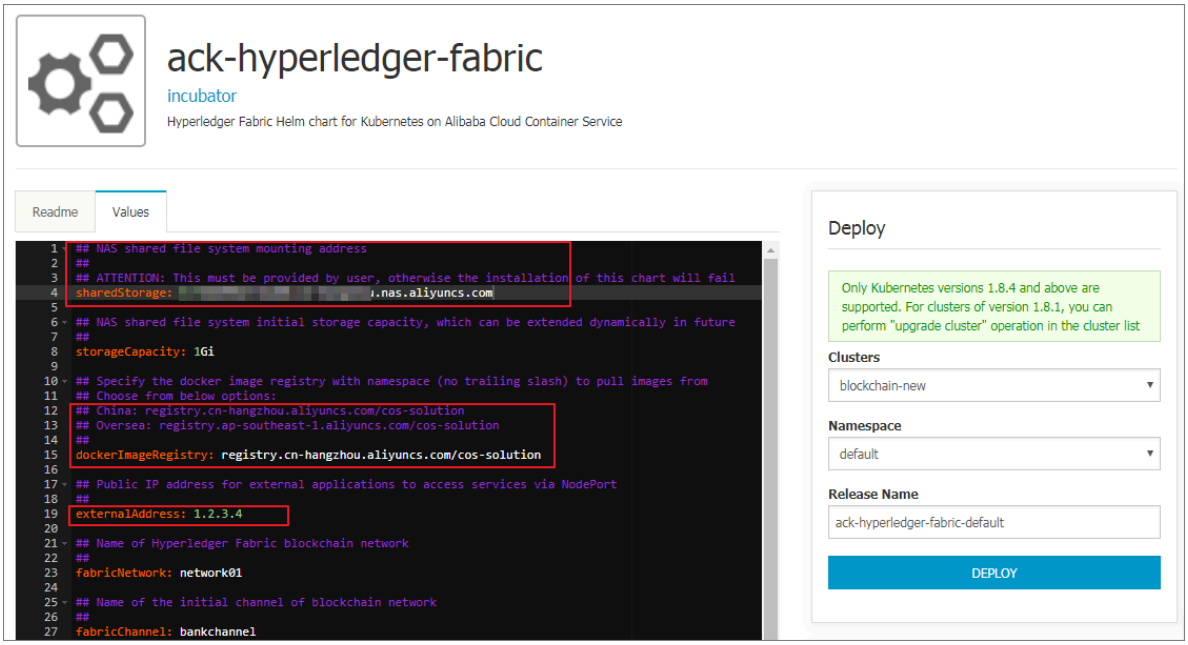
1. Under Kubernetes, click Store > App Catalog in the left-side navigation pane. Click ack-hyperledger-fabric , and select the region.



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



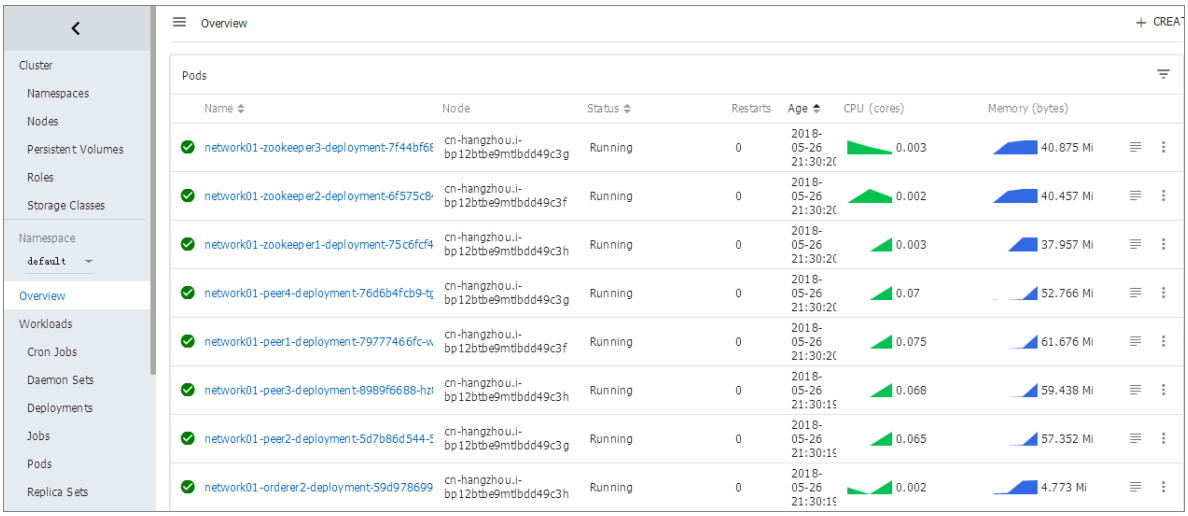
3. Click the Values tab. View or modify the corresponding deployment parameters.
   - `sharedStorage`: Create a file system and add a NAS file system mount address recorded in the mount point (required, otherwise the deployment fails).
   - `dockerImageRegistry`: Enter the container image repository address in the annotations as the value according to the region in which the blockchain network is to be deployed (inside or outside China).
   - `externalAddress`: Enter the EIP bound to the worker node to generate the connection profile.



4. Click Deploy.

> **Note:**
>
> If `sharedStorage` is not configured, the `spec.NFS.error for server: required value` error is reported during the deployment process. When the error occurs, you must delete the corresponding publication name and then enter the `sharedStorage` parameter values, and redeploy.

5. Enter the cluster dashboard. Check the status of the pods related to the blockchain network. Wait until the statuses of all the pods become Running.



The Kubernetes service of the container dashboard also supports deploying the blockchain network by using Helm. For more information, see Configure and deploy blockchain network.

Step 5. Test blockchain network by using CLI

1. On a master node of the Kubernetes cluster, run the following command to enter the CLI container.

```
kubectl exec -it <fabricNetwork>-fabric-cli bash
```

2. Run the following command to start the CLI test.

```
./cli-test.sh
```

3. During the test, press any key to continue.
4. If no errors occur during the test and the following words are displayed, it indicates that the test is successfully completed.

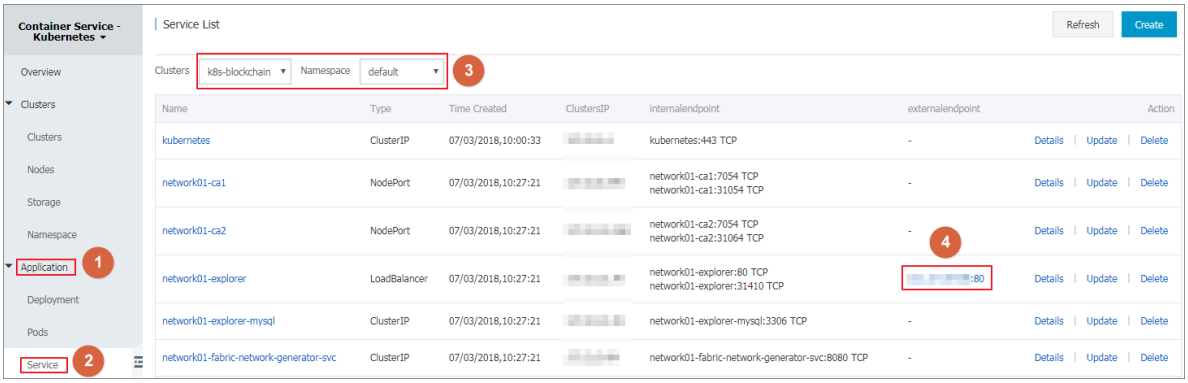
```
==================== Query on PEER4 on channel 'bankchannel' is successful ====================
Press any key to continue...
==================== All GOOD, End-2-End execution completed ====================
```

Step 6. Access blockchain explorer

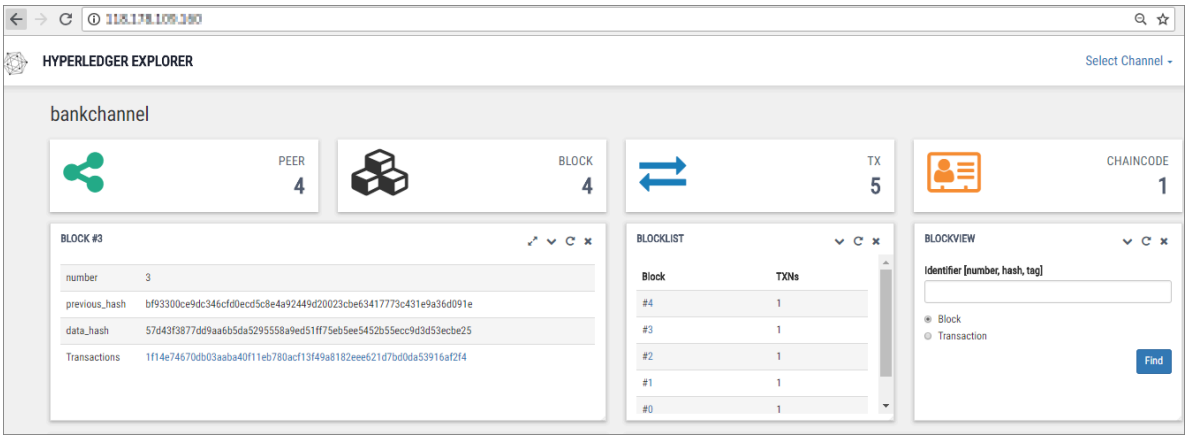
1. Log on to the Container Service console.

2. Under the Kubernetes, click Application > Services in the left-side navigation pane. Select the target cluster and namespace, find the <Network name>-explorer service and access external endpoint.

   You can also run the `kubectl get svc` command on the master node of the Kubernetes cluster, or log on to the Container Service console. Then, go to the Kubernetes clusters, and click Services in the left-side navigation pane. Then view the EXTERNAL-IP (external endpoint) of the <Network name>-explorer service.

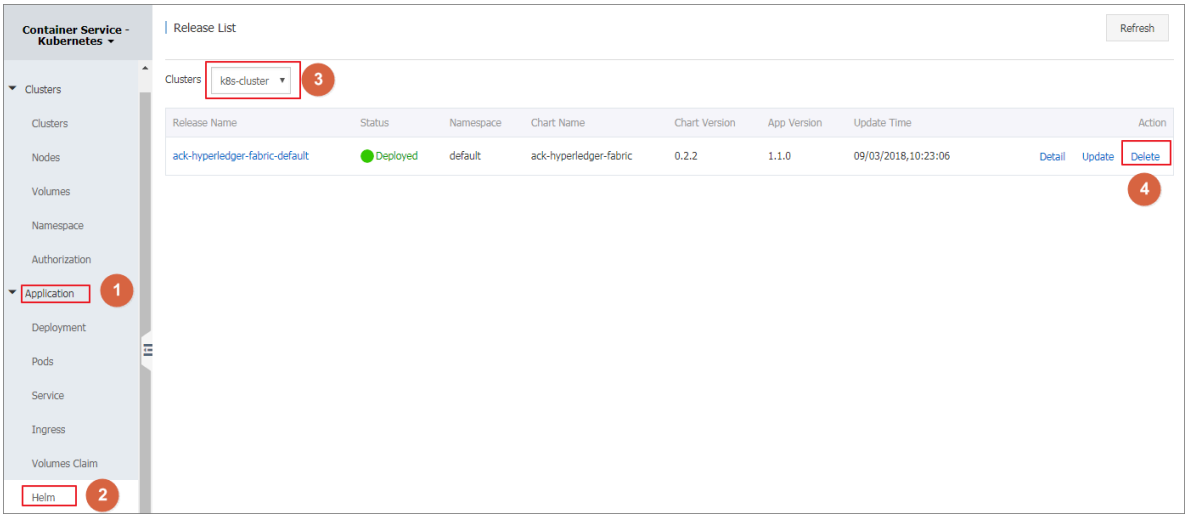


3. Access the external endpoint in the browser.

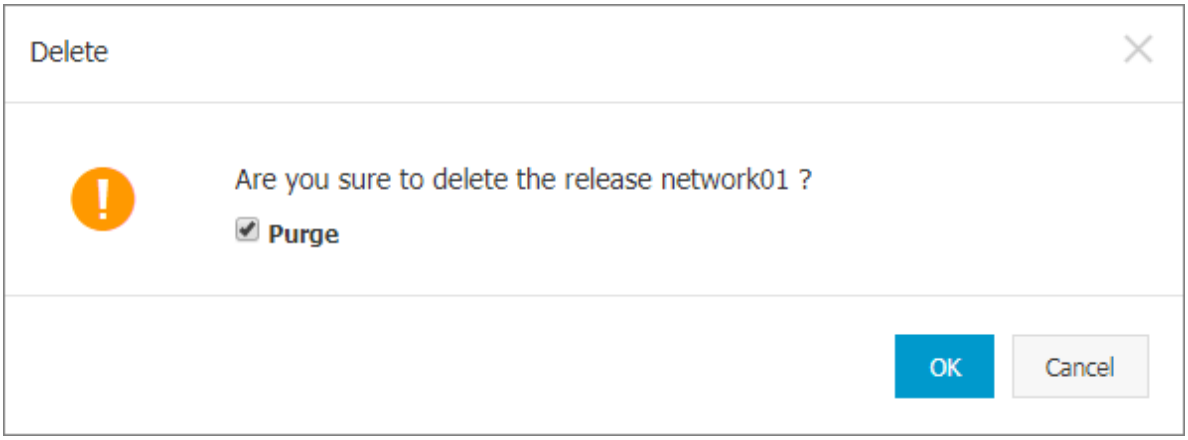


Step 7. Delete blockchain network

1. Under the Kubernetes, click Application > Helm in the left-side navigation pane. Select the cluster from the Clusters drop-down list. Click Delete at the right of the release name of the blockchain network.



2. Click OK in the displayed dialog box.



Then, the environment preparations, configuration and deployment, test, and deletion of the blockchain network are complete. For further development and testing, you can repeat the steps of blockchain configuration and deployment, blockchain testing, and blockchain deletion. You can also customize the network environment according to product instructions as needed.

1.4 Environment preparations

Before using the Alibaba Cloud Container Service blockchain solution, complete the corresponding environment preparations, which mainly includes:

· Create a Kubernetes cluster

- Bind an Elastic IP (EIP) to a worker node
- Create a file system and add a mount point

This document introduces how to prepare the environment.

Create a Kubernetes cluster

The blockchain solution deployment is based on a Kubernetes cluster built by ECS. For how to create a Kubernetes cluster, see Create a Kubernetes cluster. When creating a Kubernetes cluster, you must complete the following key configurations to make sure the blockchain solution can be deployed successfully.

- Region: The Kubernetes cluster and the NAS file system must be in the same region. Make sure the selected region supports both the Kubernetes cluster and the NAS file system (intersection of both). For the list of regions that support the NAS file system, see Products > Storage & CDN > NAS > File System List.
- Network type: Select Virtual Private Cloud (VPC).
- SSH login: For ease of management, select the Enable SSH access for Internet check box.
- Node configurations: We recommend that you use the default settings (such as three master nodes and three worker nodes). You can also configure the nodes as per your needs. The numbers of softwares, services, and containers deployed by the blockchain network are large. Therefore, make sure the cluster resources can meet the requirements.

Click Create Cluster. Wait several minutes (depends on the number of ECS instances), until the Kubernetes cluster and the ECS instances are created.

After the Kubernetes cluster is created, click Manage at the right of the created cluster on the Cluster List page. On the Basic Information page > Connection Information section, record the Master node SSH IP address, which is an Internet address and used as the external address.

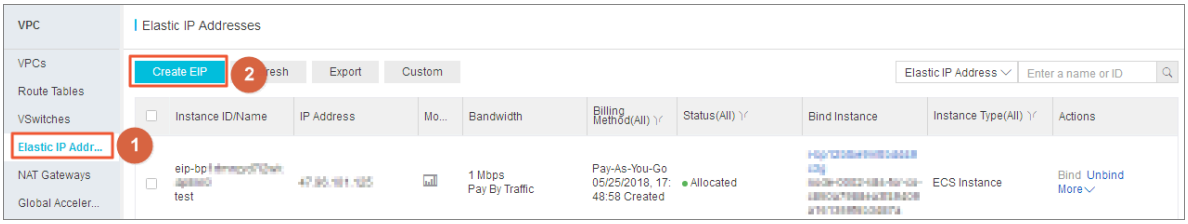
Click Kubernetes > Clusters > > Nodes in the left-side navigation pane. Record the IP addresses (intranet addresses) of the cluster nodes (both master nodes and worker nodes) for later usage.

Bind an EIP to a worker node

This section introduces the prerequisite for the access to the blockchain network from outside the cluster, which creates and binds an EIP to a worker node of the Kubernetes cluster.

Procedure

1. Log on to the Elastic IP Address (EIP) console.
2. Click Elastic IP Addresses in the left-side navigation pane.
3. Click Create EIP.



4. Select the region in which the Kubernetes cluster resides. Complete the other configurations as per your needs. Click Buy Now.
5. After activating the EIP, wait until the purchased EIP is displayed in the list and then click Bind at the right of the EIP.
6. Select ECS Instance from the Instance Type drop-down list and select a worker node (the instance name begins with node, instead of master) from the ECS Instance drop-down list. Click OK.
7. After binding the EIP to the worker node, record the IP address of the purchased EIP on the Elastic IP Addresses page.

Create a file system and add a mount point

In the blockchain solution, a file system is mainly used to store and distribute the certificates, keys, and configurations of the blockchain and persistently store the data of the blockchain master nodes.

> **Note:**
> The usage of a file system only applies to the development and test phases of blockchain applications and solutions. For deployment in the production environment, contact us to further discuss specific business and technical needs and determine the most suitable method.

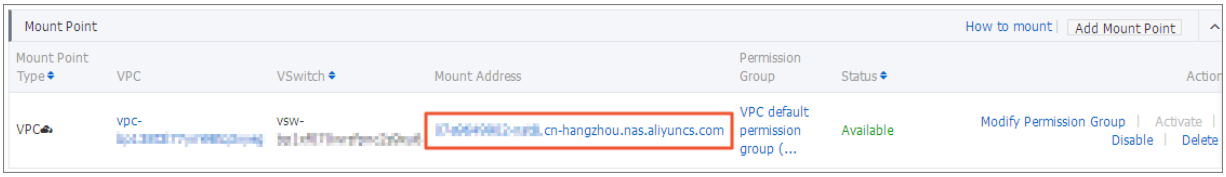
For how to create a file system, see Mount a file system on an ECS instance running Linux. When creating a file system, you must complete the following key configurations to make sure the blockchain solution can be deployed successfully.

- Region: Select the region in which the container cluster resides.
- Protocol Type: Select NFS.

For more information, see ../../SP_111/DNnas1858274/EN-US_TP_18691.dita#concept_60431_zh.

After adding a mount point, click Manage at the right of the file system.

Record the Mount Address of the mount point.



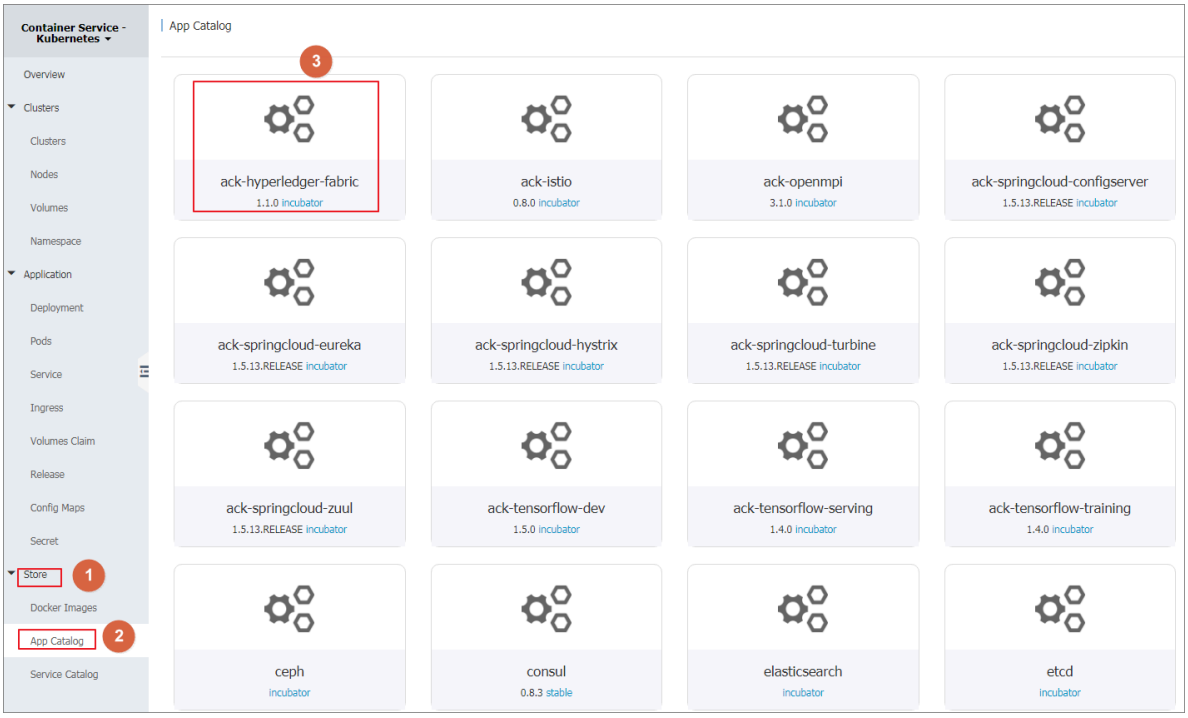
1.5 Access blockchain solution homepage

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. Under Kubernetes, click 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.

1.6 Configure and deploy blockchain network

After completing the environment preparations, you must configure and deploy the blockchain network. The blockchain network is a blockchain running environment , which is based on Hyperledger Fabric and consists 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 documentation.

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 the graphical interface or by using the Helm commands. 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 configuration process. To customize a blockchain, you can set parameters by referring to the following parameter description.

Parameter	Description
sharedStorage	The mount address of the NAS file system. Parameters that must be provided to create a blockchain network, otherwise the creation fails.
storageCapacity	The initial size of the Persistent Volume Claim (PVC) of the NAS in Kubernetes. NAS supports dynamic scaling. The default size is 1G.

Parameter	Description
dockerImageRegistry	The Docker image repository URL: The image repository of the blockchain solution. Select an image repository based on the region in which the Kubernetes cluster resides. · Inside China: registry.cn-hangzhou.aliyuncs.com/cos-solution · Outside China: registry.ap-southeast-1.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.
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 access address (mandatory). To use an application or a management and monitoring tool outside the container cluster to access the blockchain network, the public IP address of a node in the Kubernetes cluster or the public IP address of the Server Load Balancer instance of the Kubernetes cluster must be used as the external access address. For more information about the configuration method, see Bind an EIP to a worker node.
ordererDomain	Orderer domain: The Orderer domain in Hyperledger Fabric. Set this parameter based on actual requirements.

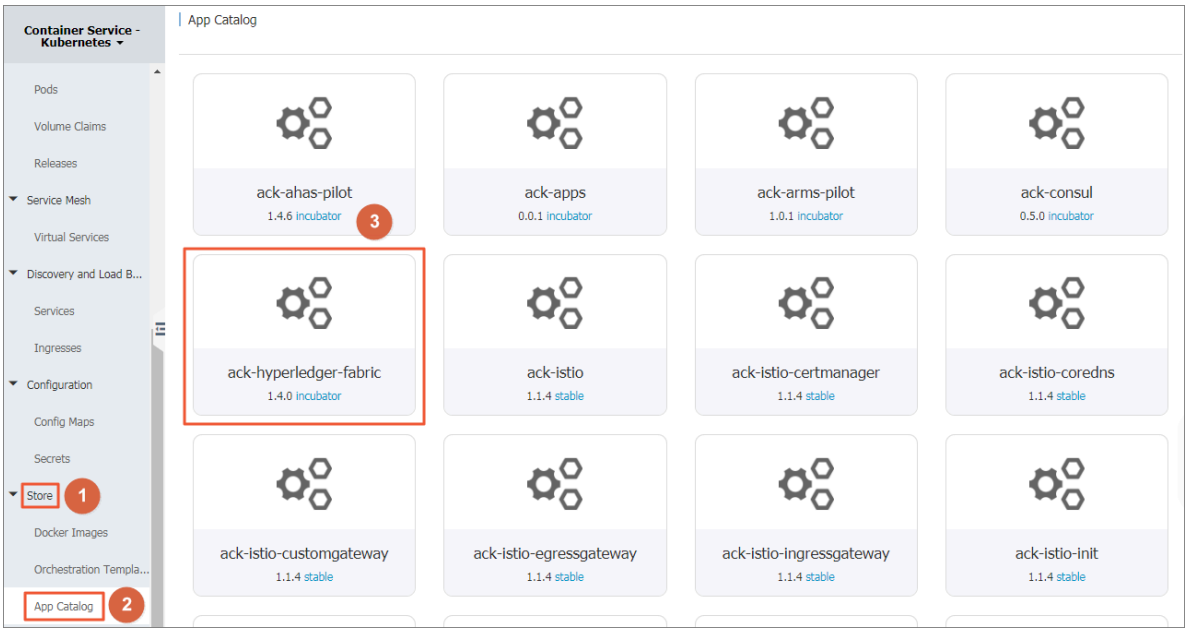
Parameter	Description
ordererNum	Number of Orderer nodes: This parameter applies to the Orderer services of the Kafka type, not the Solo type. Set this parameter to the number of Orderer nodes to be deployed. To modify the parameter value, you must modify the value of ordererExternalPortList at the same 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.
peerDomain	Peer domain: The peer domain in Hyperledger Fabric. Set this parameter based on actual requirements.
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 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.
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.

Parameter	Description
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.
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.
imagePullPolicy	Image pull policy: This is a Kubernetes parameter and is generally used for development and test.
hyperledgerFabricVersion	Hyperledger Fabric version: Currently, 1.4.0 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, 0.4.14 is supported, which corresponds to the Hyperledger Fabric 1.4.0, and no configuration is required.
explorer.enabled	Whether 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. The default value is true.

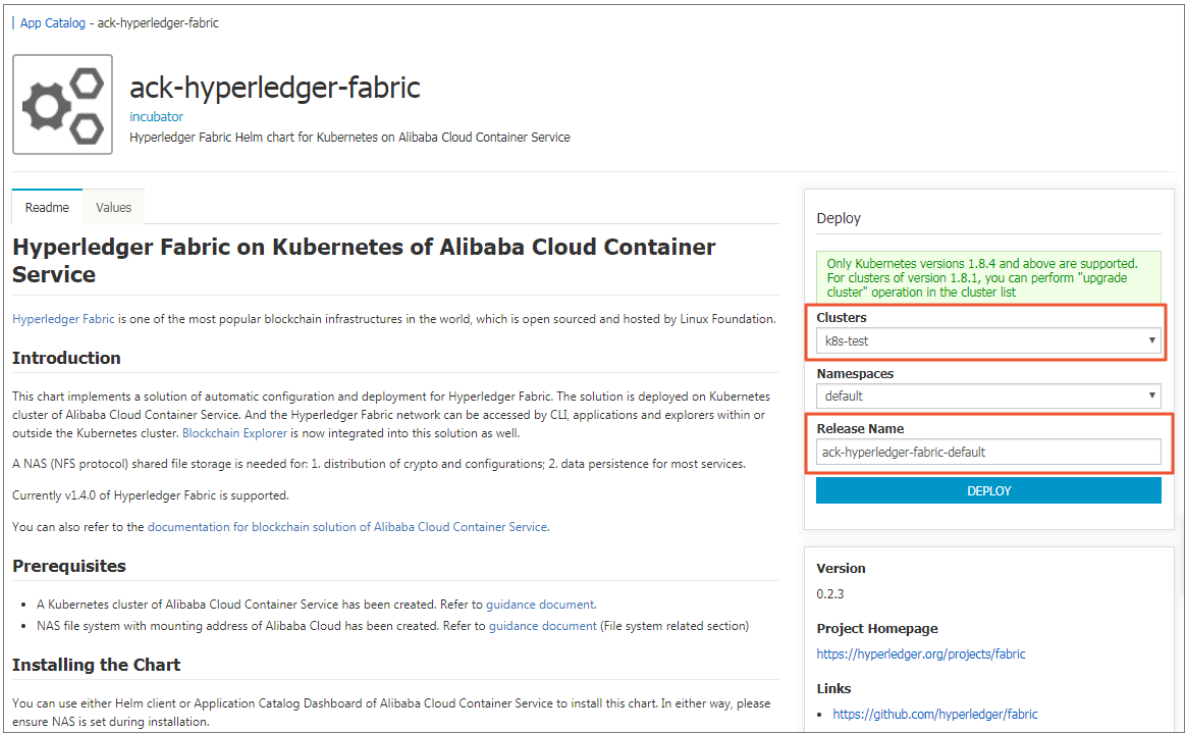
Parameter	Description
logService.enabled	Whether to enable the support for Alibaba Cloud Log Service. The default value is false. For more information about Alibaba Cloud Log Service, see Use Log Service to collect Kubernetes cluster logs.
logService.region	Specify the region in which the Log Service project resides if logService.enabled is set to true. Set the parameter value as per your needs. For more information about Alibaba Cloud Log Service, see Use Log Service to collect Kubernetes cluster logs.
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 Use Log Service to collect Kubernetes cluster logs.
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 Use Log Service to collect Kubernetes cluster logs.
logLevel	Hyperledger Fabric log levels for different types of nodes (peer, orderer, couchdb).. Optional values are: CRITICAL | ERROR | WARNING | NOTICE | INFO | DEBUG, the default value is INFO.

Deploy blockchain network in Container Service console

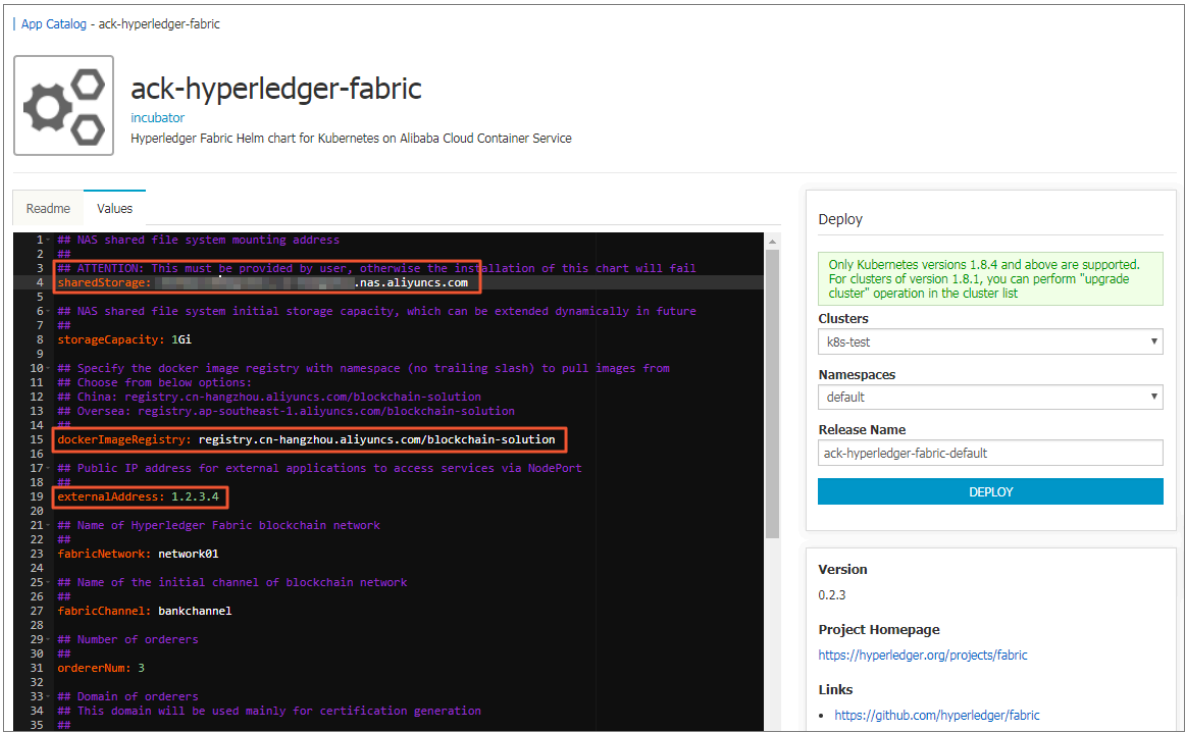
1. Log on to the Container Service console. In the left-side navigation pane under Kubernetes, choose Store > App Catalog. Then, click ack-hyperledger-fabric



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



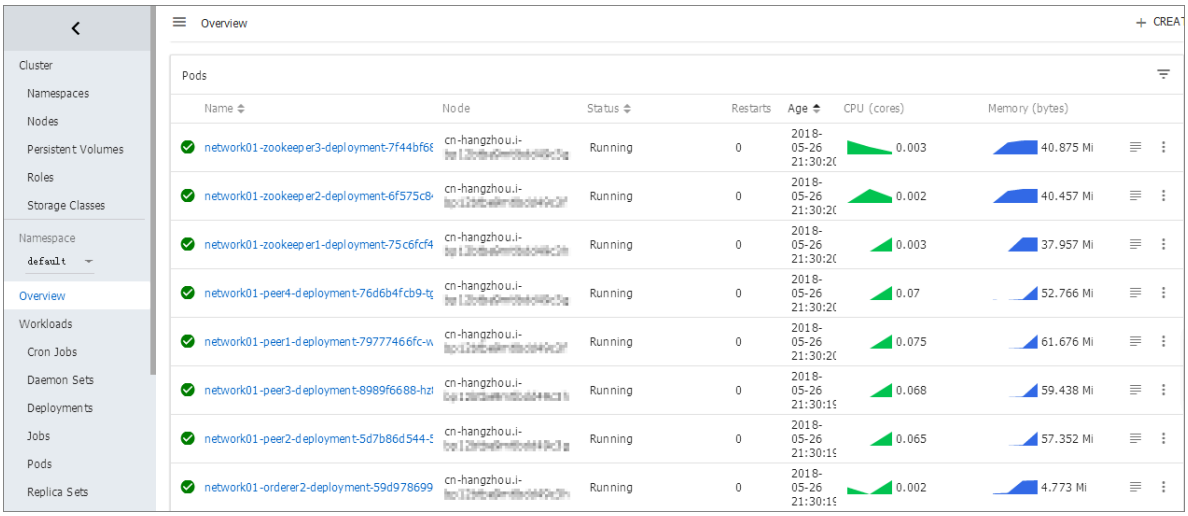
3. Click the Values tab. View or modify the corresponding deployment parameters.
   - sharedStorage: Enter the NAS file system mount address during the environment preparation (required, otherwise the deployment fails).
   - dockerImageRegistry: Enter the container image repository address in the annotations as the value according to the region in which the blockchain network is to be deployed (inside or outside China).
   - externalAddress: Enter the EIP bound to the worker node to generate the connection profile.



4. Click Deploy.

> **Note:**
> If sharedStorage is not configured, the spec.NFS.error for server: required value error is reported during the deployment process. When the error occurs, you must delete the corresponding publication and then enter the sharedStorage parameter values, and redeploy.

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 Simplify Kubernetes application deployment by using Helm.

1. Use SSH to log on to the master node in a Kubernetes cluster with the root account. Enter the password configured when the Kubernetes cluster is created.
2. Run Helm commands to deploy the blockchain network.
   - To deploy the blockchain network by using the default parameter configurations, run the following command:

     ```
     helm install --name blockchain-network01 --set "
     sharedStorage=987a6543bc-abc12.cn-hangzhou.nas.
     aliyuncs.com" 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 a blockchain network to the specified namespace (for example, network01), run the following command:

     ```
     helm install --namespace network01 --name blockchain-
     network01 --set "sharedStorage=987a6543bc-abc12.cn-
     ```

```
hangzhou . nas . aliyuncs . com "  incubator / ack - hyperledge
r - fabric
```

Wherein, `-- namespace` indicates the name of the 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  "
fabricChan  nel = mychannel , sharedStor  age = 987a6543bc -
abc12 . cn - hangzhou . nas . aliyuncs . com "  incubator / ack -
hyperledge  r - 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 - hyperledge  r - 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:

```
#  sample   values   yaml
 sharedStor  age :  987a6543bc - abc12 . cn - hangzhou . nas .
 aliyuncs . com
 fabricNetw  ork :  network01
 fabricChan  nel :  tradechann  el
 orgNum :  3
 ordererNum :  4
 ordererDom  ain :  shop
 peerDomain :  shop
 externalAd  dress :  11 . 22 . 33 . 44
 caExternal  PortList : [" 31054 ", " 31064 ", " 31074 "]
 ordererExt  ernalPortL  ist : [" 31050 ", " 31060 ", " 31070 ", "
 31080 "]
 peerExtern  alGrpcPort  List : [" 31051 ", " 31061 ", " 31071 ",
 " 31081 ", " 31091 ", " 31101 "]
```

Check if the 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 If the status is `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 certificat e 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 approaches, 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.

Then, use Kubernetes-related commands, such as `kubectl describe pod`, `kubectl logs`, and `kubectl get pod -o yaml`, to view the 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. 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. For example:

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

Finally, for problems or errors related to Hyperledger Fabric, use Hyperledger Fabric official documentation, StackOverFlow, or Google/Bing/Baidu for relevant information or solutions.

1.7 View blockchain network logs

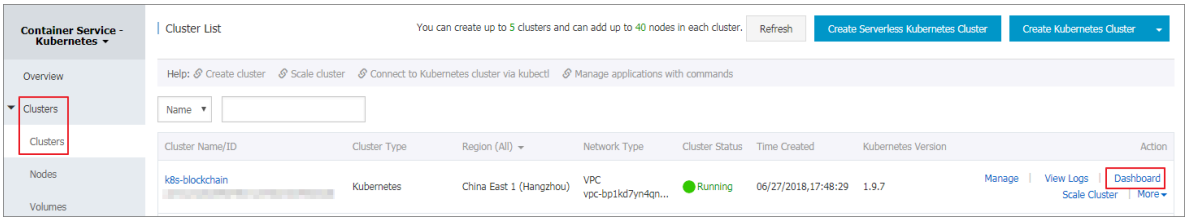
Logs generated when the blockchain network is running are output as container logs of Peer, Orderer, CA, Kafka, and ZooKeeper node types. The Container Service blockchain solution allows you to view these logs in the Container Service console, by using Kubernetes commands, or by means of Alibaba Cloud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

View logs in Container Service console

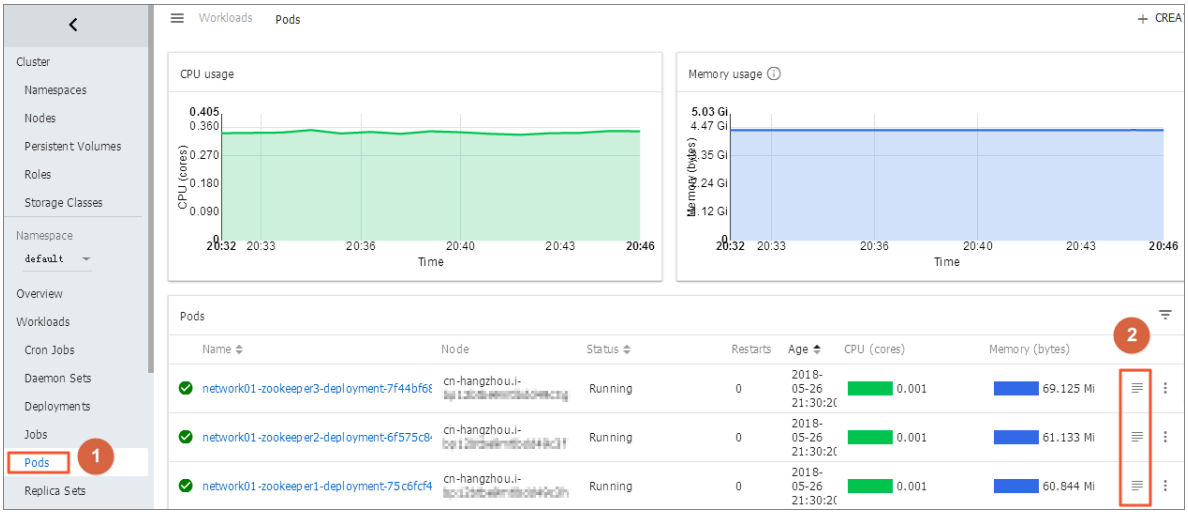
The Container Service console allows you to conveniently view logs on GUI. After deploying the blockchain network, you can follow these steps to view the container logs of the corresponding node.

Procedure

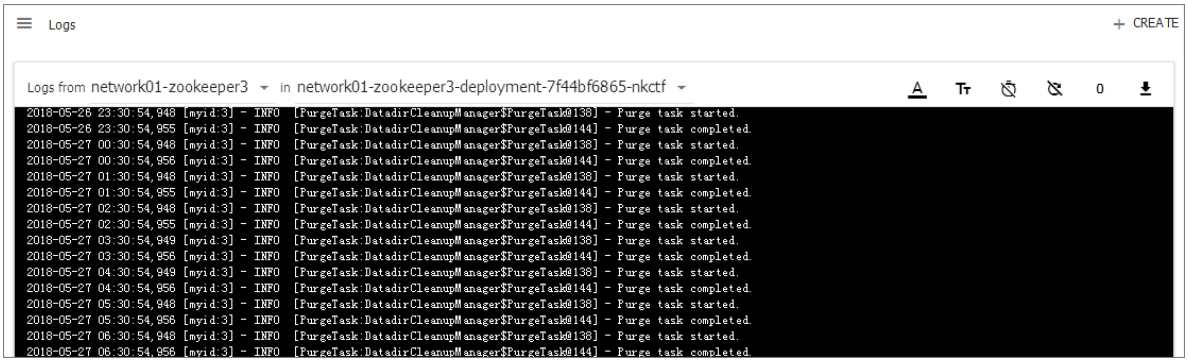
1. Log on to the Container Service console.
2. Under Kubernetess, click Clusters in the left-side navigation pane. Click Dashboard at the right of the cluster.



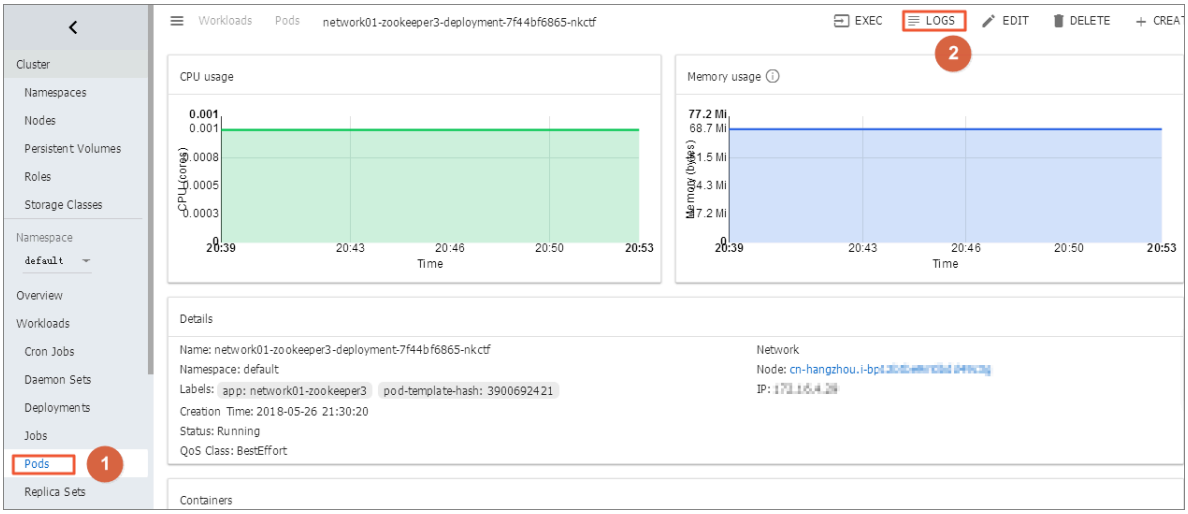
3. Click Pods in the left-side navigation pane and then click the Logs icon.



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

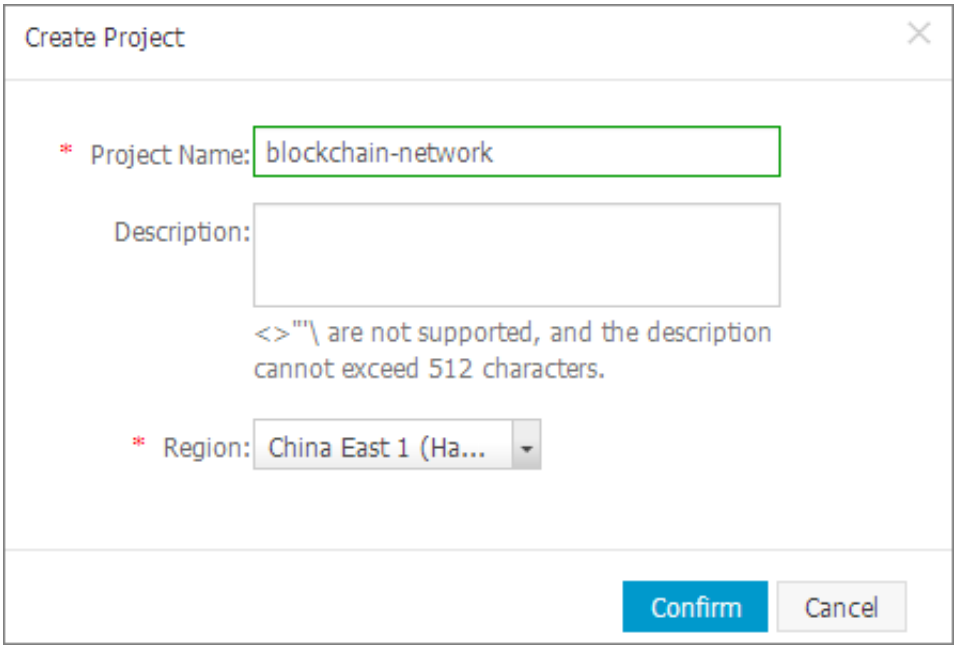
The log function of the Container Service console and the Kubernetes commands can basically meet the common requirements of log viewing. However, enterprise level requirements may include advanced functions such as log storage, real-time query and analysis, alarm, and visualized report. Then, you can 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 Application log management.

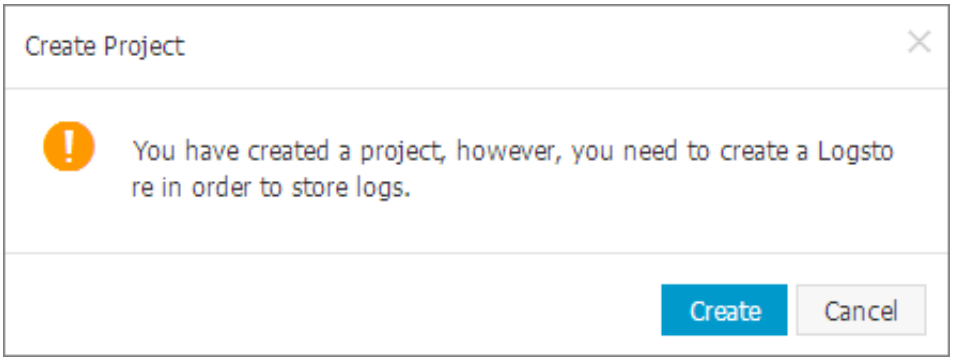
You may be charged by using Alibaba Cloud Log Service. For more information, see Log Service billing method.

Procedure

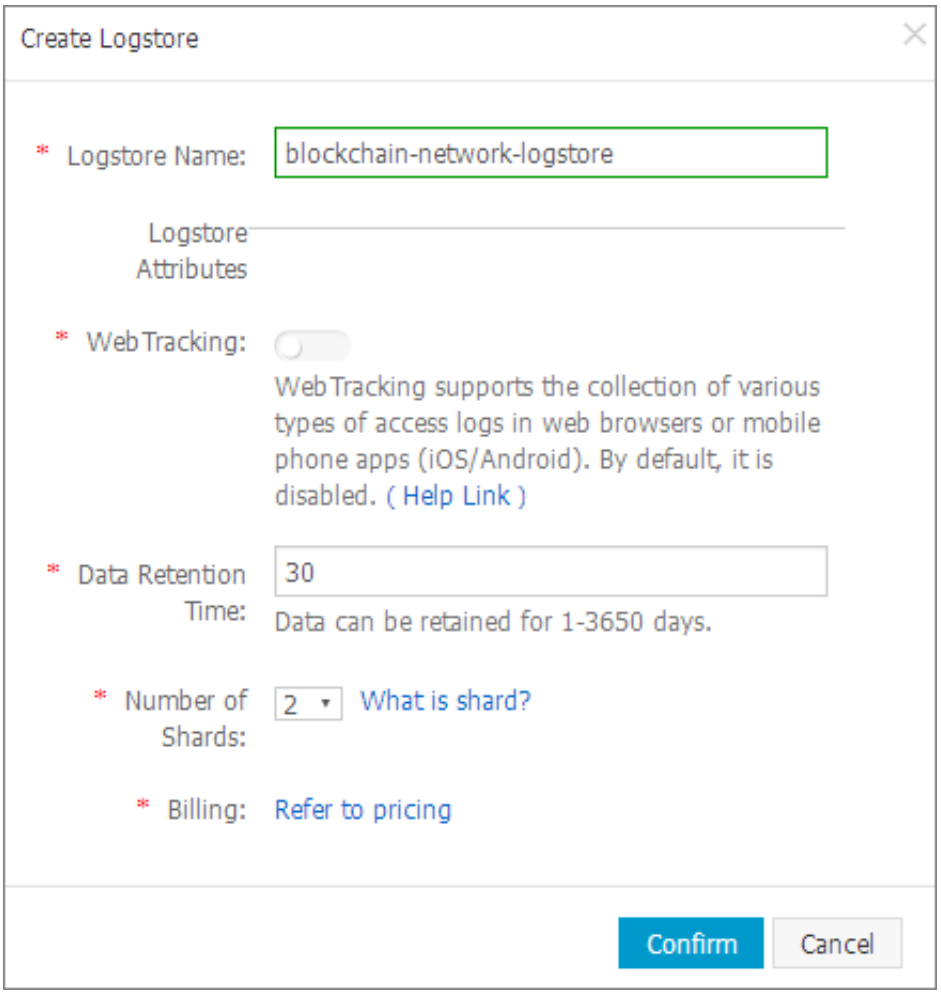
1. Log on to the Log Service console and activate Log Service as instructed.
2. Click Project Management in the left-side navigation pane, and click Create Projectin the upper-right corner.
3. Enter the Project Name, and select the Region in which the blockchain network resides. Then, click Confirm.



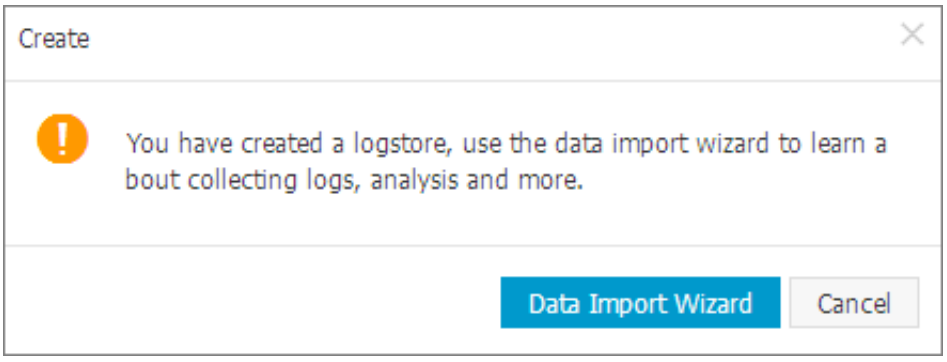
4. Click Create in the displayed dialog box to create the Logstore.



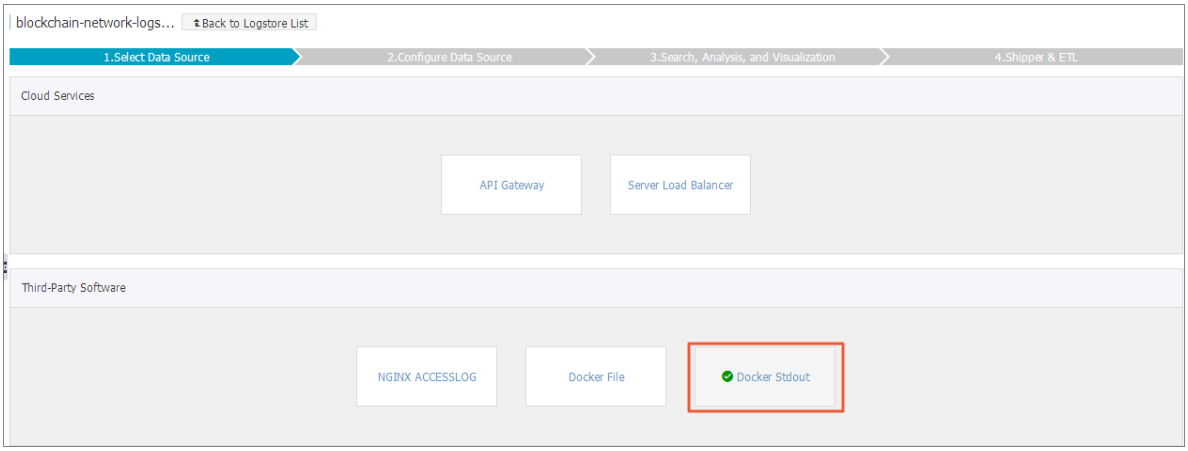
5. The Create Logstore dialog box appears. Enter the Logstore name. Complete the other configurations as per your needs. Then, click Confirm.



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



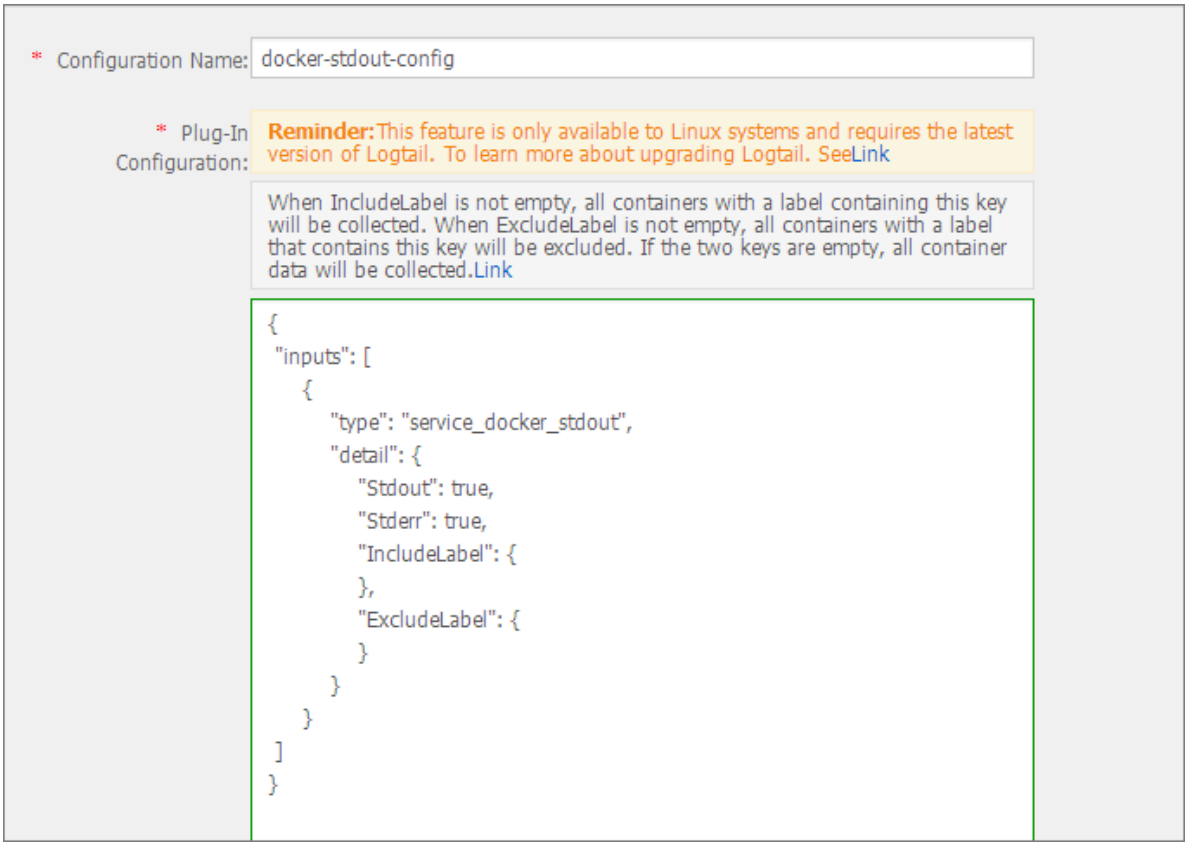
7. Select Docker Stdout under Third-Party Software. Then, click Next.



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_do  cker_stdou  t ",
          " detail ": {
              " Stdout ":  true ,
              " Stderr ":  true ,
              " IncludeLab  el ": {
              },
              " ExcludeLab  el ": {
              }
          }
      }
  ]
```

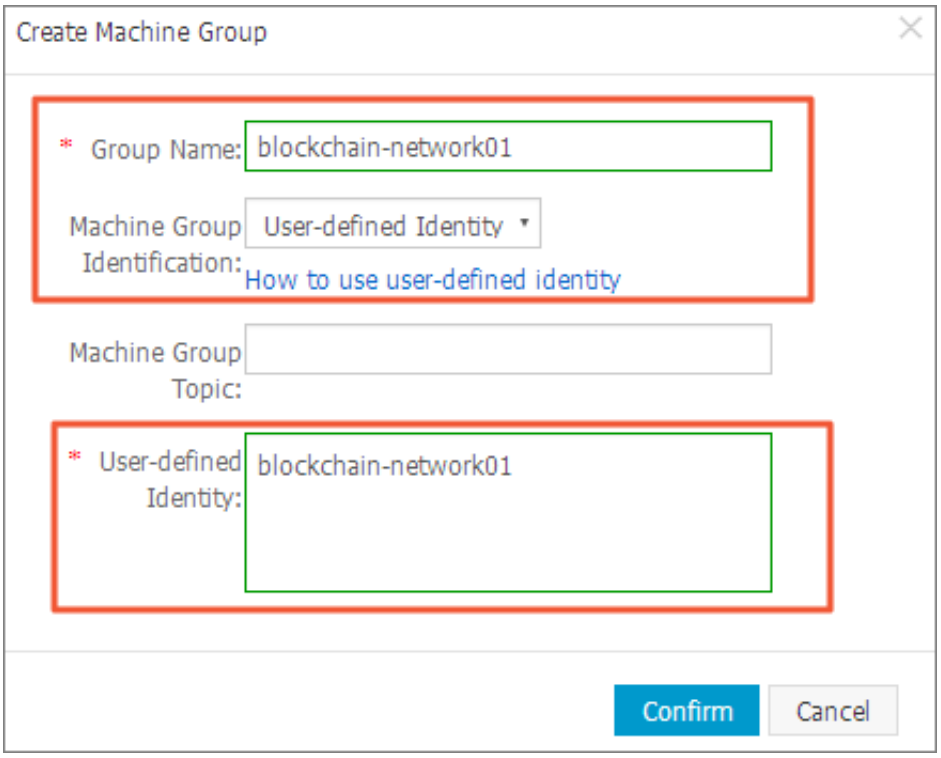
```
}
```



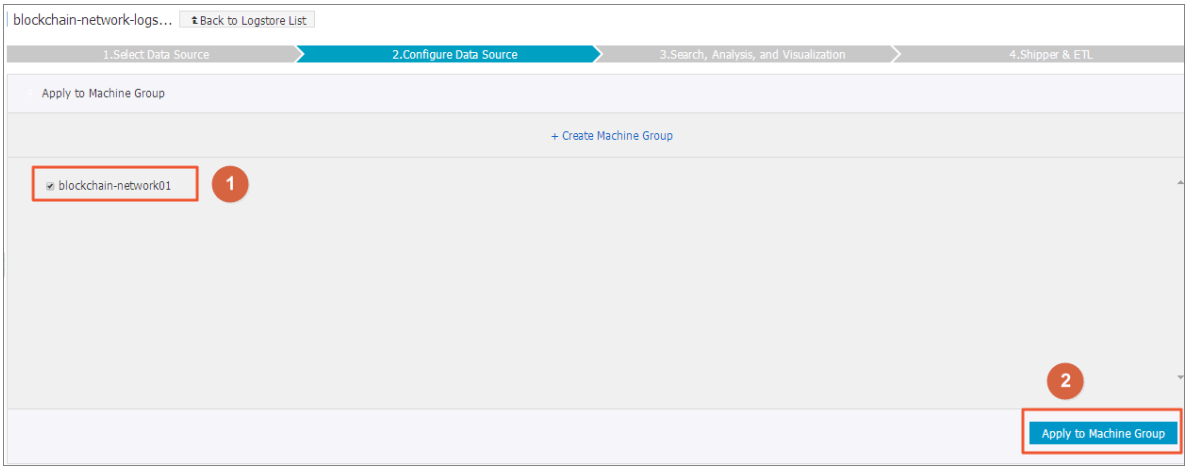
9. **Click Create Machine Group.**

   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

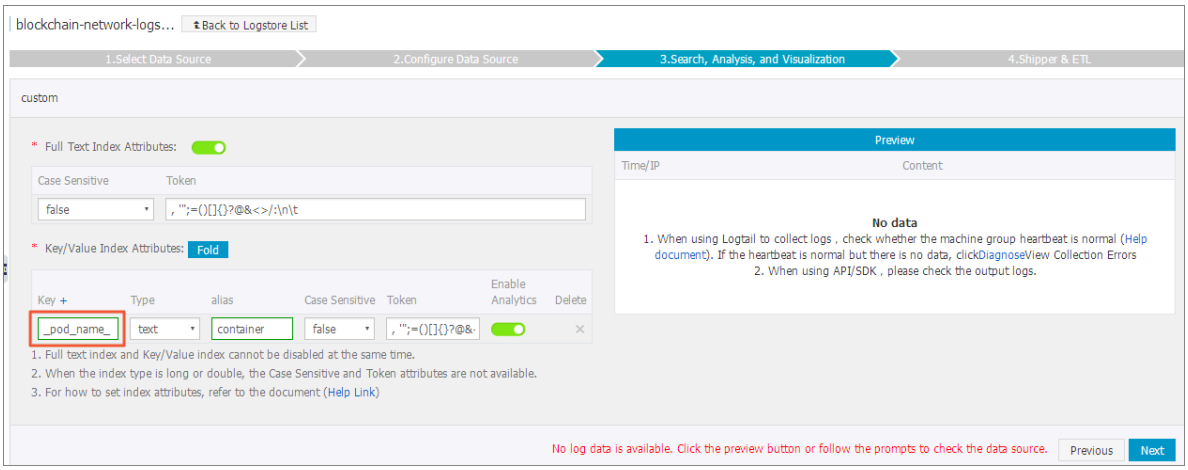
same content as that in the Group Name field. Then, click Confirm. The example is as follows:



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

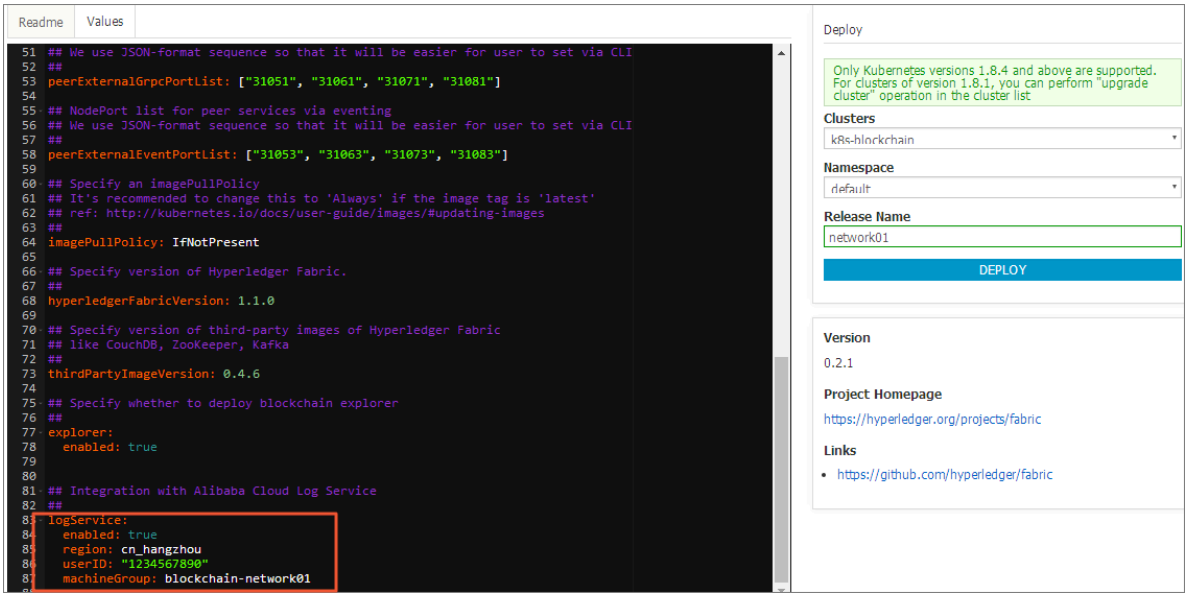


11. Add the key (for example, _pod_name_) for creating the index as per your needs. After completing the configurations, click Next.



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.

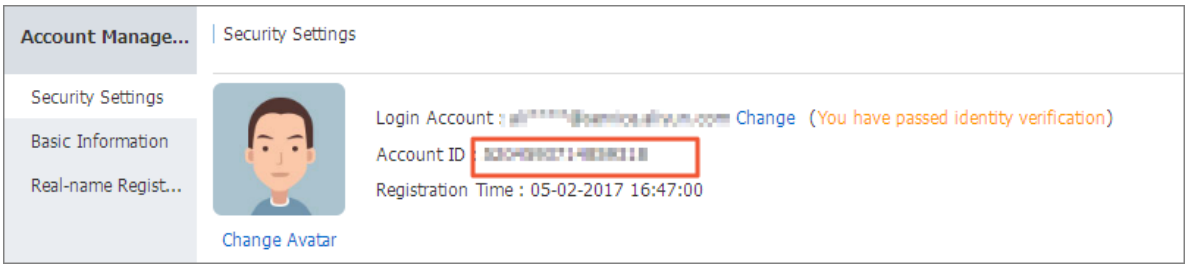


Set the parameter `enabled` to true, which indicates to enable Log Service. Set the parameter `machineGroup` to the user-defined identity configured in the machine group. In this example, it is blockchain-network01.

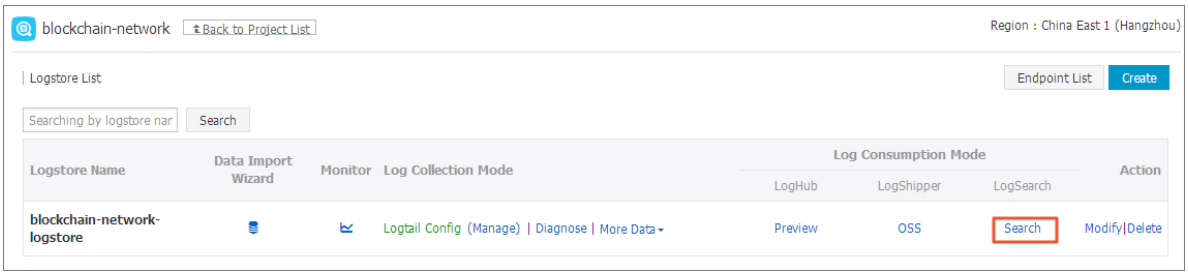
To set the parameter region, see Linux to search for the corresponding installation commands and the region ID. For example, `cn_hangzhou` indicates to write

logs from the Alibaba Cloud intranet in the region Hangzhou to Log Service and no Internet bandwidth is consumed.

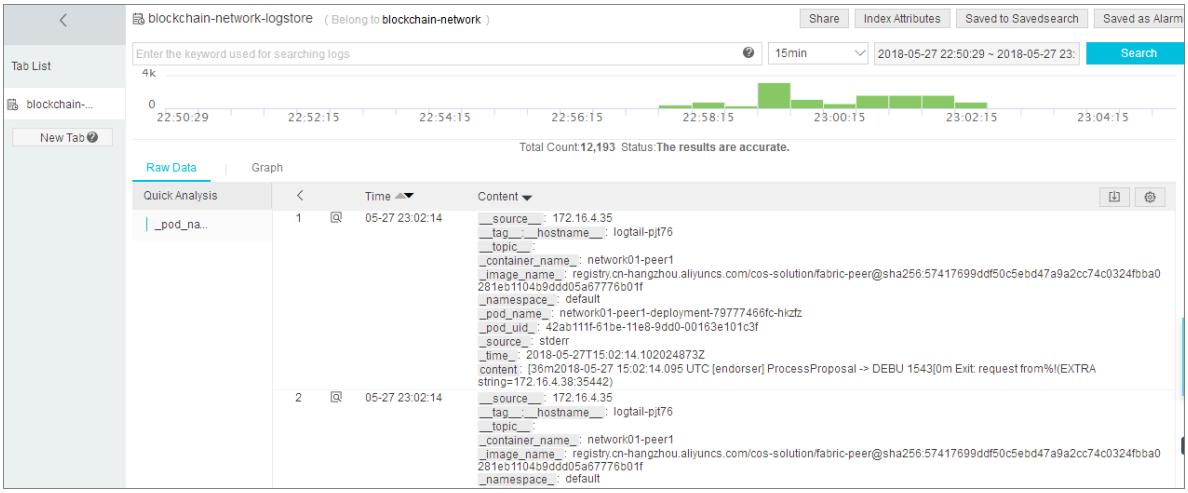
To set the parameter userID, see the following figure.



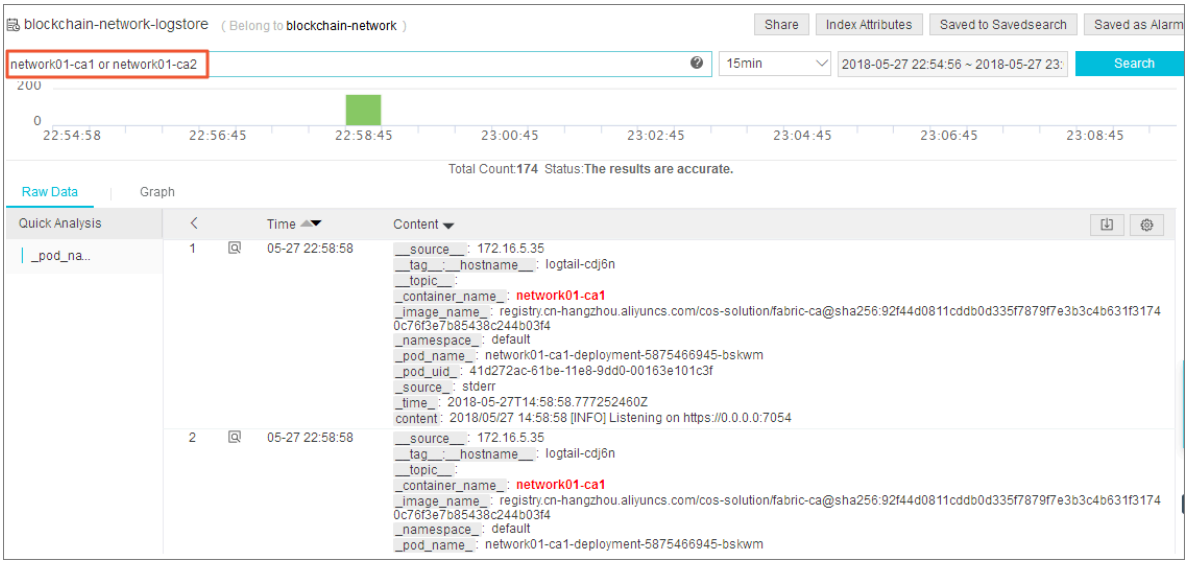
14. Log on to the Log Service console. Click the project name and then click Search at the right of the Logstore.



The blockchain network log example is as follows:



15. Log Service supports complex queries. For more information about the query syntax and other advanced functions of Log Service, see ../../SP_7/DNSLS11850791/EN-US_TP_13099.dita#concept_tnd_1jq_zdb.



1.8 Overview about blockchain network access

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 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 in one click (including certificates, keys and blockchain network configuration files necessary for access to blockchain services) to accelerate the development and testing processes of blockchain applications and management and monitoring tools.

- Configure Internet IP address and external port
- Access blockchain network by using CLI
- #unique_25
- Access blockchain network by using administration and monitoring tools

1.9 Configure Internet IP address and external port

Prerequisites

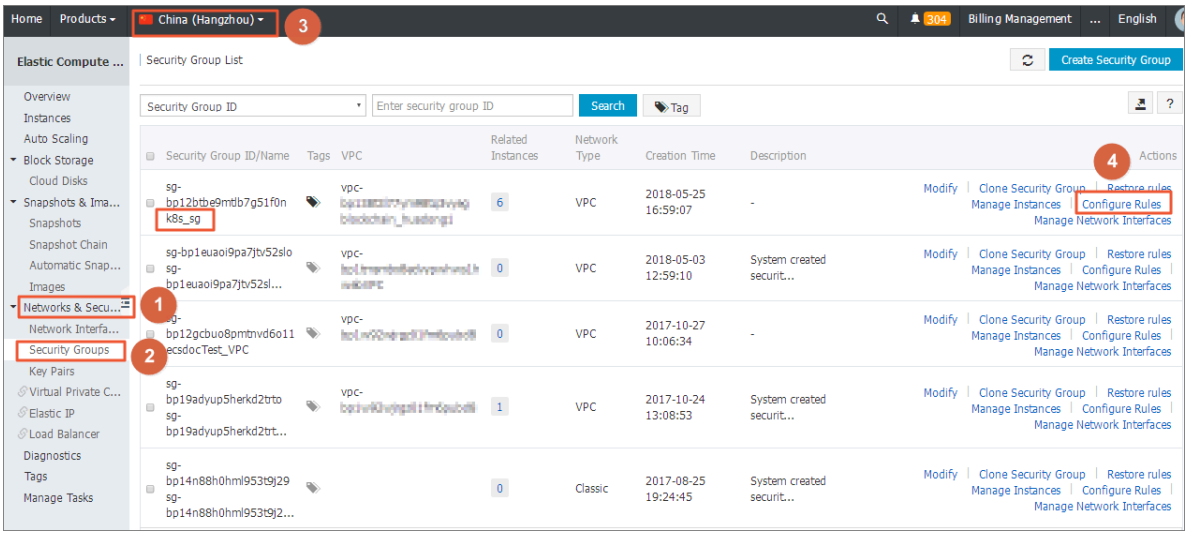
The following prerequisites must be met before making the blockchain network available to administration and monitoring tools and applications outside the container cluster.

- Possess the externally accessible Internet address. You can either bind an EIP to a worker node or create a Server Load Balancer instance (add a worker node on the backend server) and use its Internet address. Environment preparations introduces how to bind an EIP to a worker node, see Bind an EIP to a worker node.
- Allows the inbound access from the NodePort in the external port list by configuring the ECS security group rules.

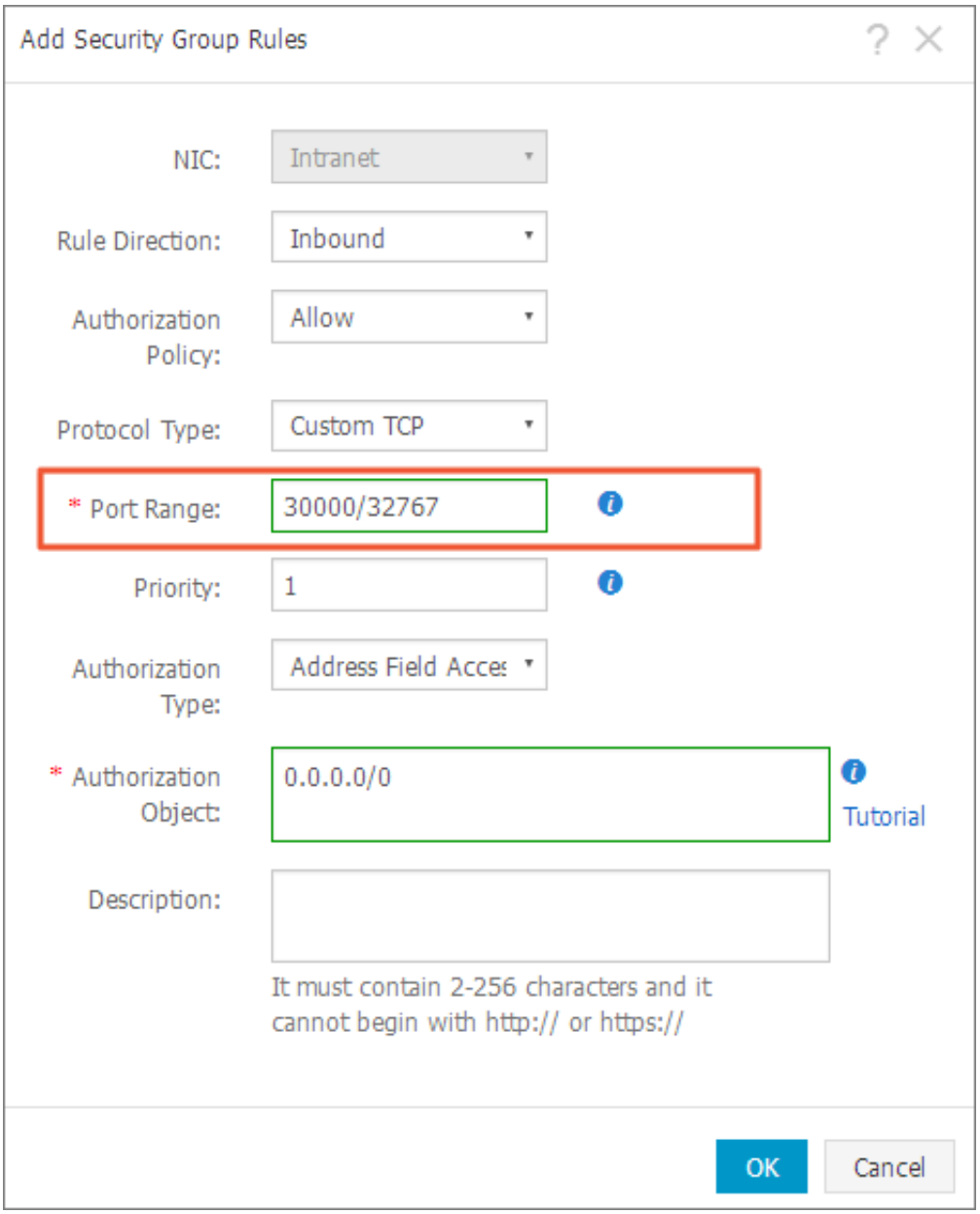
Procedure

1. Log on to the ECS console .
2. Click Networks & Security > Security Groupsin the left-side navigation pane.
3. Select the region in which your Kubernetes cluster resides.

4. The security group name of the Kubernetes cluster contains k8s_sg. Click Configure Rules at the right of the security group.



5. If no security group rules under the Inbound tab meets the requirements, click Add Security Group Rules. The Add Security Group Rules dialog box appears. Complete the configurations. Complete the configurations.



- Select Inbound from the Rule Direction drop-down list.
- Enter the suitable Port Range according to the external port NodePort range of the blockchain network.
- Enter the suitable address range in the Authorization Object field based on the actual access requirements.

6. Click OK. The security group rule is successfully added to the list and takes effect immediately.

1.10 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 `kubectl exec -it <fabricNetwork>-fabric-cli -n <namespace name> bash` command in the 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 `kubectl exec -it -n <namespace name> <fabricNetwork>-fabric-cli bash` command to enter the CLI container, for example, `kubectl exec -it -n network01 <fabricNetwork>-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. Choose to run only one type of sample for each blockchain network.

1.11 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 / AliyunCont ainerServi ce /
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.

1.12 Access blockchain network by using administration and monitoring tools

Prerequisites

- Create the Alibaba Cloud blockchain network by using the Container Service blockchain solution. For more information, see Configure and deploy blockchain network.
- Access the blockchain network by using applications or CLI and complete the end-to-end test. For more information, see Access blockchain network by using CLI or #unique_25.

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 administration and monitoring tools based on Hyperledger Fabric SDKs.

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

In this example, deploy the administration and monitoring tool outside Alibaba Cloud container clusters. The provided administration and monitoring tool is developed and adapted based on Hyperledger Explorer . When the blockchain network is deployed, the Hyperledger Explorer is deployed to the Kubernetes cluster by default.

You can use the administration and monitoring tool provided in this example, the official version of Hyperledger Explorer, self-developed administration and monitoring tool, or third-party administration and monitoring tool, and adapt the tool by referring to the source codes of the administration and monitoring tool provided in this example to access the Alibaba Cloud Container Service blockchain network.

Adapt the existing blockchain administration and monitoring tool as follows:

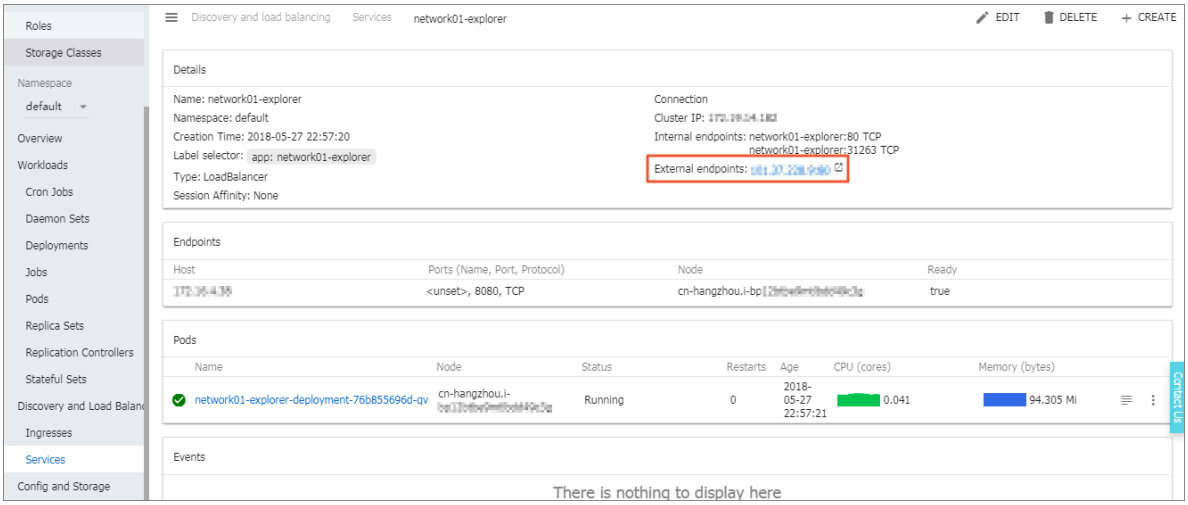
- Directly use the script download-from-fabric-network.sh that is provided in the sample code to automatically download blockchain network configurations from the newly deployed blockchain network with one click, including certificates, keys

, and blockchain network configuration files (basically the config.json and network-config.yaml).

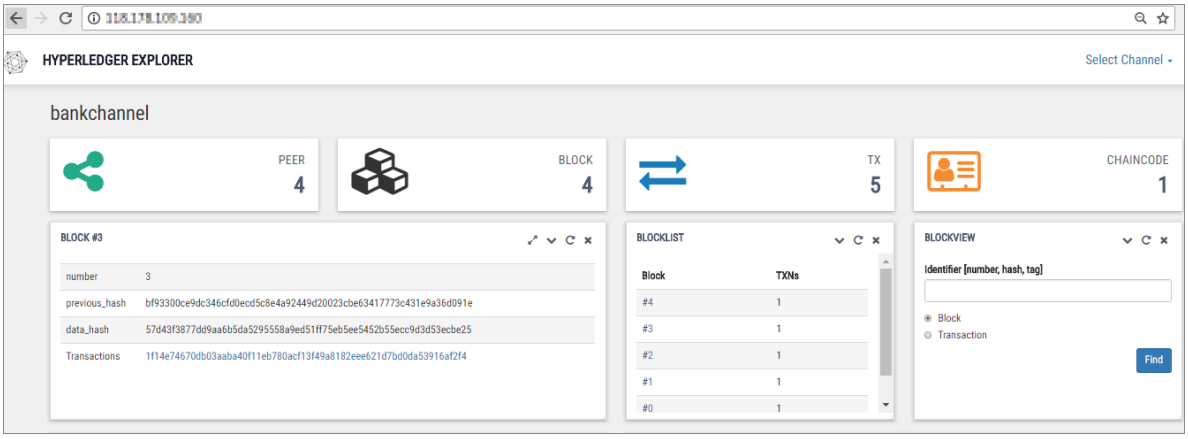
- Make sure that the blockchain administration and monitoring tool can be correctly loaded to the preceding blockchain network configurations.
- If the blockchain administration and monitoring tool directly uses the channel name, external address, node name, or domain name of the blockchain, replace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

Procedure

1. Run the `kubectl get svc` command on a master node of the Kubernetes cluster. You can also log on to the Container Service console, click Services in the left-side navigation pane, and check the external endpoint of the service <network name>-explorer.



2. Access the external endpoint in the browser.



Note:

Hyperledger Explorer is still in the project incubation phase, and the functionality remains to be improved. You can follow its official project progress to get further functions and version updates.

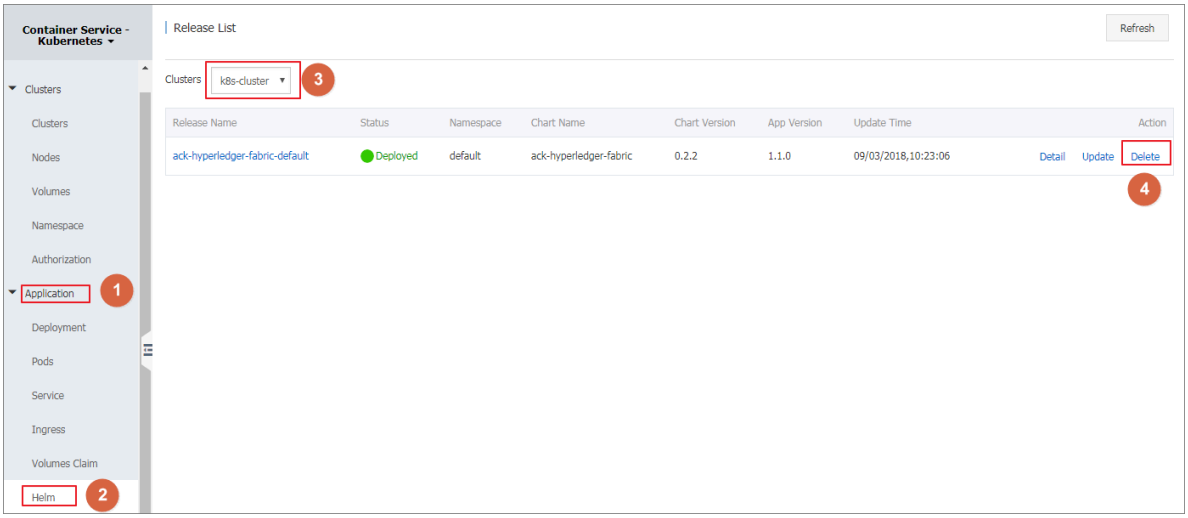
1.13 Clean up blockchain environment

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.

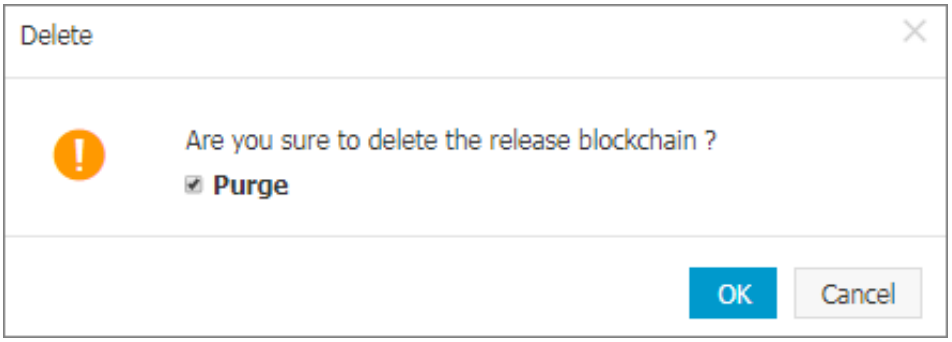
Delete blockchain network in Container Service console

1. Log on to the Container Service console.

2. Under Kubernetes, click Application > Helm. Select the cluster from the Clusters drop-down list. Click Delete at the right of the release name of the blockchain network.



3. Click OK in the displayed dialog box.



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. Execute the `helm delete --purge <Helm release name of the blockchain network>` command to delete the blockchain network. For example, `helm delete --purge blockchain-network01`.

Depending on the number of nodes on the blockchain network, wait for several minutes until the helm delete command is completed and the result is returned. So far, the services and containers corresponding to all nodes on the blockchain network have been deleted from the Kubernetes cluster. The chaincode 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.

To access or clean up the blockchain data catalog, you can use the following command example to mount a NAS file system to ECS.

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

1.14 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  " sharedStor  age
= 029bb489d2 - ikw80 . cn - hangzhou . nas . aliyuncs . com "
incubator / ack - hyperledge  r - 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.