User Guide
V2.2.14

Kubernetes CSI

Issue	01
Date	2021-09-29

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

About This Document

Intended Audience

This document is intended for:

- Technical support engineers
- O&M engineers
- Engineers with basic knowledge of storage and Kubernetes

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
DANGER	Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.
WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Supplements the important information in the main text. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Contents

1 Overview

This document describes how to deploy and use the Kubernetes CSI plug-in so that Huawei enterprise and distributed storage devices provide persistent volume storage capabilities for Kubernetes.

2 Environmental Requirements

- Kubernetes has been deployed and is running properly.
- A Huawei storage device is running properly.
- Tools for scanning disks and mounting files must be installed on all worker hosts of Kubernetes in advance. If containers and services cannot run properly due to lack of system tools, view logs by referring to **9.1 Viewing Log Information** and install the tools on the hosts.

Table 2-1 Version mappings between Kubernetes and enterprise storage products

Kubernetes Version	Enterprise Storage Product Version
1.13/1.14/1.15/1.16/1.17/1.18/1.19/1.20/1.21	OceanStor Dorado V6 6.0.0/6.0.1/6.1.0/6.1.2 OceanStor Dorado V3 V300R002 OceanStor F V5/V5 V500R007/V500R007 Kunpeng OceanStor F V3/V3 V300R006

Table 2-2 Version mappings between Kubernetes and distributed storage products

Kubernetes Version	Distributed Storage Product Version
1.13/1.14/1.15/1.16/1.17/1.18/1.19/1.20/1.21	FusionStorage V100R006C30 FusionStorage Block 8.0.0/8.0.1 OceanStor Pacific series 8.1.0

Table 2-3 Features supported by Huawei CSI (√: supported; x: not supported)

Feature	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.20	1.21
Create PVC	√	√	√	√	√	√	√	√	√

Feature	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.20	1.21
Delete PVC	√	√	√	√	√	√	√	√	√
Create POD	√	√	√	√	√	√	√	√	√
Delete POD	√	√	√	√	√	√	√	√	√
Offline Resize	x	x	x	√	√	√	√	√	√
Online Resize	x	x	x	√	√	√	√	√	√
Create Snapshot	x	x	x	x	√	√	√	√	√
Delete Snapshot	x	x	x	x	√	√	√	√	√
Restore	x	x	x	x	√	√	√	√	√
Clone	x	x	x	x	√	√	√	√	√

3 Restrictions

This section describes the restrictions on connecting CSI to storage devices.

Table 3-1 Restrictions

Scenario	Restriction	Supported Storage	Remarks
PersistentVolume-Claim (PVC) access mode	**ReadWriteOnce**: SAN/NAS **ReadWriteMany**: NAS **ReadWriteOnly**: SAN/NAS If SAN needs to use the **ReadWriteMany** mode, the Pod service must ensure data consistency.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6, OceanStor Pacific series	
PVC	A maximum of 100 items can be created or deleted in a batch.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6, OceanStor Pacific series	The maximum number of concurrent RESTful requests is 100.

Scenario	Restriction	Supported Storage	Remarks
Pod	A maximum of 100 items can be created or deleted in a batch.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6, OceanStor Pacific series	
Snapshot	N/A	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6	
Creating a PVC using a snapshot	Snapshot restoration, that is, creating a PVC using the snapshot instead of rolling back to the original PVC using the snapshot.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5	
Expanding the capacity of a PVC	• Storage resources can be expanded but cannot be reduced. • PVCs in RWO mode do not support capacity expansion.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6	
Cloning a PVC	The StorageClasses of the source and target PVCs must be the same.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5	

Scenario	Restriction	Supported Storage	Remarks
HyperMetro	• PVCs and Pods can be created only when both HyperMetro storage systems are normal. If a single storage system is faulty, only the services that have been delivered are normal and new services cannot be delivered. If both storage systems are faulty, contact Huawei technical support engineers. • HyperMetro and remote replication cannot be configured in the same StorageClass.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6 NAS: OceanStor V3/V5	
Remote replication	HyperMetro and remote replication cannot be configured in the same StorageClass.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6 NAS: OceanStor V3/V5, OceanStor Dorado V6	

Scenario	Restriction	Supported Storage	Remarks
Residual drive letter	Due to a node fault, containerized applications are migrated to other nodes. After the node recovers, residual drive letters exist on the node. Manually clear the residual drive letters. For details, see **9.4 After a Worker Node in the Cluster Breaks Down and Recovers, Pod Failover Is Complete but the Source Host Where the Pod Resides Has Residual Drive Letters**.	SAN: OceanStor V3/V5, OceanStor Dorado V3/ Dorado V6, FusionStorage 8.0	Condition: iSCSI/FC + Multipath

4 Installation and Deployment

4.1 Obtaining the Software Package

You can obtain Huawei Kubernetes CSI through Huawei Kubernetes CSI warehouse.

Step 1 Open a browser and enter **https://github.com/Huawei/eSDK_K8S_Plugin/releases** in the address box.

Step 2 Select the desired version package and download ***eSDK_Cloud_Storage_Plugin_*.*.***.zip**. **.*.** indicates the release version number.

Step 3 Decompress the package.

Step 4 Find the package and documents in the directory generated after the decompression.

----End

4.2 Components in the Software Package

Decompress ***eSDK_Cloud_Storage_Plugin_*.*.***.zip** to obtain the software package and sample files required for installing and using CSI. **Table 4-1** shows the software package structure.

Table 4-1 Component description

Component	Description
bin/huawei-csi	Implements the CSI API.
bin/secretGenerate	Encrypts plaintext passwords and produces **secret** objects.
bin/secretUpdate	Encrypts plaintext passwords and updates **secret** objects.
yamls	Collection of .yaml sample files used in subsequent deployment.

4.3 Creating a Huawei CSI Image

Huawei CSI runs as a container. Currently, Huawei CSI provides only a binary package (**bin/huawei-csi**) which cannot be used directly. Therefore, you need to create a CSI image based on the binary file to start the Huawei CSI service.

Prerequisites

A Linux host with Docker installed is available, and the host can access the Internet (only used to download the image package).

Procedure

Step 1 Log in to the Linux host.

Step 2 Run the **mkdir image** command to create a directory (for example, **image**) on the host.

```
# mkdir image
```

Step 3 Run the **cd image** command to access the **image** directory.

```
# cd image
```

Step 4 Copy the huawei-csi component to the **image** directory.

Step 5 Run the **vi Dockerfile** command to create a file named **Dockerfile**.

```
# vi Dockerfile
```

Step 6 Press **I** or **Insert** to enter the editing mode and enter the following information in the **Dockerfile** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
FROM busybox:stable-glibc

ADD ["huawei-csi", "/"]
RUN ["chmod", "+x", "/huawei-csi"]

ENTRYPOINT ["/huawei-csi"]
```

NOTICE

busybox:stable-glibc indicates the basic image and its tag. It is only an example. Replace it based on site requirements.

Step 7 Run the **docker build -f Dockerfile -t huawei-csi:2.2.13 .** command to create an image.

```
# docker build -f Dockerfile -t huawei-csi:2.2.13 .
```

NOTE

2.2.13 indicates the plug-in version number corresponding to the software package name. It is only an example. Replace it based on site requirements. If the same image already exists in the environment, use **docker image rm** *<image-id>*.

Step 8 Run the **docker image ls | grep huawei-csi** command to check whether the image is created. If the following information is displayed, it is created.

```
# docker image ls | grep huawei-csi
huawei-csi                    2.2.13              c8b5726118ac        About a minute ago   39 MB
```

Step 9 Run the **docker save huawei-csi:2.2.13 -o huawei-csi.tar** command to export the image.

```
# docker save huawei-csi:2.2.13 -o huawei-csi.tar
```

NOTE

2.2.13 indicates the plug-in version number corresponding to the software package name. It is only an example. Replace it based on site requirements.

Step 10 Run the **scp huawei-csi.tar** *<user>***@***<ip>***:/***<path>** command to copy the **huawei-csi.tar** image file to all worker nodes in the Kubernetes cluster and enter the password as prompted.

```
# scp huawei-csi.tar <user>@<ip>:/<path>
```

- *<user>*: user name for logging in to a worker node in the Kubernetes cluster.
- *<ip>*: IP address for logging in to a worker node in the Kubernetes cluster.
- *<path>*: name of the folder to be copied to a worker node in the Kubernetes cluster.

Step 11 Log in to a worker node in the Kubernetes cluster and run the **docker load -i huawei-csi.tar** command to import the image.

```
# docker load -i huawei-csi.tar
```

Step 12 After the import is complete, run the **docker image ls | grep huawei-csi** command to check whether the import is successful. If the following information is displayed, the import is successful.

```
# docker image ls | grep huawei-csi
huawei-csi                    2.2.13          c8b5726118ac    10 minutes ago   39MB
```

Step 13 Repeat **Step 10** to **Step 12** to import the image to all worker nodes in the Kubernetes cluster.

----End

4.4 Configuring Host Multipathing

4.4.1 Installing the Multipathing Tool Package

This section describes how to install the multipathing tool package.

Prerequisites

Ensure that the worker nodes in the Kubernetes cluster can access the Internet (only used to download the multipathing tool package).

Precautions

The host multipathing service needs to be configured only when storage devices are connected over an iSCSI or FC network.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to a worker node in the Kubernetes cluster through the management IP address.

Step 2 Install the multipathing tool package based on the operating system (OS).

- CentOS:
```
yum install -y device-mapper-multipath
```
- Ubuntu:
```
apt-get install -y multipath-tools
apt-get install -y multipath-tools-boot
```
- SUSE:
```
zypper install -y multipath-tools
```

Step 3 Enable the host multipathing service.

- CentOS:
```
/sbin/mpathconf --enable
systemctl start multipathd.service
systemctl enable multipathd.service
systemctl restart multipathd.service
```
- Ubuntu:
```
systemctl restart multipath-tools.service
```
- SUSE:
```
systemctl restart multipath-tools.service
chkconfig multipathd on
```

Step 4 Repeat **Step 1** to **Step 3** to install the multipathing tool on all worker nodes.

----End

4.4.2 Configuring the Multipathing Service

Multipathing is configured to improve the link reliability of LUNs on SAN storage. If multipathing is incorrectly configured, I/O errors will occur when a single link is faulty. As a result, the file systems or disks in the containers managed by the Kubernetes cluster are read-only or faulty, affecting I/O delivery.

Precautions

The multipathing service needs to be configured only when storage devices are connected over an iSCSI or FC network.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to a worker node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi /etc/multipath.conf** command to modify the **multipath.conf** file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

NOTICE

Load balancing mode: During service read and write, the I/O paths from a host to all controllers on a storage device are the same. For details, see **Configuring Multipathing** > **Concepts** in ***Huawei SAN Storage Host Connectivity Guide for Red Hat***.

Local preferred mode: When a host delivers I/Os to controllers, the storage device with better performance is accessed because the service link distances from storage devices are different. For details, see **Configuring Multipathing** > **Concepts** in ***Huawei SAN Storage Host Connectivity Guide for Red Hat***.

- If enterprise storage and the load balancing mode are used, you are advised to add the following content to the **devices** field in the multipathing configuration file (**/etc/multipath.conf**). For details, see ***OceanStor Dorado Host Connectivity Guide for Red Hat***.

```
devices {
    device {
            vendor                  "HUAWEI"
            product                 "XSG1"
            path_grouping_policy        multibus
            path_checker            tur
            prio                    const
            path_selector           "service-time 0"
            failback                immediate
            no_path_retry           15
        }
}
```

- If enterprise storage and the local preferred mode are used, you are advised to add the following content to the **devices** field in the multipathing configuration file (**/etc/multipath.conf**). For details, see ***Huawei SAN Storage Host Connectivity Guide for Red Hat***.

```
devices {
    device {
            vendor              "HUAWEI"
            product             "XSG1"
            path_grouping_policy    group_by_prio
            path_checker            tur
            prio                alua
            path_selector           "round-robin 0"
            failback            immediate
            no_path_retry           15
        }
}
```

- If distributed storage is used, you are advised to add the following content to the **devices** field in the multipathing configuration file (**/etc/multipath.conf**). The configuration varies according to the OS. For details, see **Configuring Multipathing for an Application Server (Red Hat or CentOS)** in ***FusionStorage 8.0.1 Block Storage Basic Service Configuration Guide 08***.

```
devices {
    device {
            vendor                  "Huawei"
            product                 "VBS fileIO"
            path_grouping_policy        multibus
            path_checker            tur
```



```
        prio                    const
        path_selector           "service-time 0"
        failback                immediate
        no_path_retry           "10"
    }
}
```

Step 3 After the configuration is complete, run the following command to restart the multipathd service.

```
systemctl restart multipathd.service
```

Step 4 Repeat **Step 1** to **Step 3** to configure the multipathing service for all worker nodes.

----End

4.5 Connecting to Enterprise Storage

This section describes how to connect the huawei-csi plug-in to Huawei enterprise storage.

4.5.1 Connecting to Enterprise Storage SAN over iSCSI

Perform this operation when you want to connect to enterprise storage SAN over iSCSI.

Prerequisites

- An iSCSI client has been installed on all worker nodes of Kubernetes.
- All worker nodes of Kubernetes communicate properly with the management IP address of the storage device to be connected.
- All worker nodes of Kubernetes communicate properly with the service IP address of the storage device to be connected.
- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes. For details, see **4.4 Configuring Host Multipathing**.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-oceanstor-iscsi.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-2**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
```



```
csi.json: |
  {
      "backends": [
          {
              "storage": "oceanstor-san",
              "name": "storage",
              "urls": ["https://192.168.125.20:8088", "https://192.168.125.21:8088"],
              "pools": ["storagepool01", "storagepool02"],
              "parameters": {"protocol": "iscsi", "portals": ["192.168.125.22", "192.168.125.23"]
          }
      ]
  }
```

Table 4-2 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the enterprise storage SAN is connected, the value is fixed to **oceanstor-san**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools that support the block storage service.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:8088 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in scenarios where the iSCSI protocol is used. This parameter is mandatory.	In scenarios where the iSCSI protocol is used, set the **protocol** parameter to a fixed value: **iscsi**. Set the **portals** parameter to the iSCSI service IP addresses of the storage device. Use commas (,) to separate multiple iSCSI service IP addresses. You can log in to DeviceManager to obtain the iSCSI service IP addresses. ● OceanStor V3/V5 series: To obtain the IP address of an Ethernet port on a specified storage device, choose **Provisioning** > **Port** > **Ethernet Ports** on DeviceManager. ● OceanStor Dorado V3 series: To obtain the IP address of an Ethernet port on a specified storage device, choose **Provisioning** > **Port** > **Ethernet Ports** on DeviceManager. ● OceanStor Dorado V6 series: To obtain the IP address of a logical port on a specified storage device, choose **Services** > **Network** > **Logical Ports** on DeviceManager and obtain the IP address whose data protocol is iSCSI.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

4.5.2 Connecting to Enterprise Storage SAN over FC

Perform this operation when you want to connect to enterprise storage SAN over FC.

Restrictions

To connect to enterprise storage SAN over FC, ensure that no residual drive letter exists on the host. If any residual drive letter exists, clear the drive letter by referring to **9.4 After a Worker Node in the Cluster Breaks Down and Recovers, Pod Failover Is Complete but the Source Host Where the Pod Resides Has Residual Drive Letters**.

Prerequisites

- All worker nodes of Kubernetes communicate properly with the management IP address of the storage device to be connected.
- All worker nodes of Kubernetes can communicate with the storage device to be connected over FC.
- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes of Kubernetes. For details, see **4.4 Configuring Host Multipathing**.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-oceanstor-fc.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-3**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-san",
                "name": "storage",
                "urls": ["https://192.168.125.20:8088", "https://192.168.125.21:8088"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "fc"}
            }
        ]
    }
```


Table 4-3 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the enterprise storage SAN is connected, the value is fixed to **oceanstor-san**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools that support the block storage service.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:8088 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.
parameters	Dictionary	Variable parameters in scenarios where the FC protocol is used. This parameter is mandatory.	In scenarios where the FC protocol is used, set the **protocol** parameter to a fixed value: **fc**.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.


```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.5.3 Connecting to Enterprise Storage NAS over NFS

Perform this operation when you want to connect to enterprise storage NAS over NFS.

Prerequisites

- An NFS client tool has been installed on all worker nodes of Kubernetes.
- All worker nodes of Kubernetes communicate properly with the storage device to be connected.
- All worker nodes of Kubernetes communicate properly with the NFS logical port of the storage device to be connected.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-oceanstor-nfs.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-4**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-nas",
                "name": "storage",
                "urls": ["https://192.168.125.20:8088", "https://192.168.125.21:8088"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "nfs", "portals": ["192.168.125.22"]}
            }
        ]
    }
```


Table 4-4 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the enterprise storage NAS is connected, the value is fixed to **oceanstor-nas**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools that support the file storage service.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:8088 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.
parameters	Dictionary	Variable parameters in scenarios where the NAS protocol is used. This parameter is mandatory.	The **protocol** parameter is fixed to **nfs**. **portals**: logical port IP address or DNS zone of the storage device. Only one IP address or DNS zone can be configured.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.


```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.5.4 Connecting to Enterprise Storage SAN over NVMe over RoCE

Perform this operation when you want to connect to enterprise storage SAN over NVMe over RoCE.

Prerequisites

- All worker nodes of Kubernetes communicate properly with the management IP address of the storage device to be connected.
- All worker nodes of Kubernetes communicate properly with the service IP address of the storage device to be connected.
- An NVMe client has been installed on all worker nodes of Kubernetes.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.
- For details about the compatibility of NVMe-related storage, hosts, and multipathing software, see **https://support-open.huawei.com/en/pages/user/compatibility/support-matrix.jsf**.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-oceanstor-roce.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-5**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-san",
                "name": "storage",
                "urls": ["https://192.168.125.20:8088", "https:/192.168.125.21:8088"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "roce", "portals": ["192.168.125.22", "192.168.125.23"]}
            }
        ]
    }
```


Table 4-5 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the enterprise storage SAN is connected, the value is fixed to **oceanstor-san**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:8088 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.
parameters	Dictionary	Variable parameters in scenarios where the RoCE protocol is used. This parameter is mandatory.	In scenarios where the RoCE protocol is used, set the **protocol** parameter to a fixed value: **roce**. Set **portals** to the IP addresses of the logical ports when data protocol type of the storage device is NVMe over RoCE. Use commas (,) to separate the IP addresses.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.


```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.6 Connecting to Distributed Storage

This section describes how to connect the huawei-csi plug-in to Huawei distributed storage.

4.6.1 Connecting to Distributed Storage SAN over SCSI

Perform this operation when you want to connect to distributed storage SAN over SCSI.

Prerequisites

- The distributed storage VBS client has been installed on all worker nodes of Kubernetes.
- All worker nodes of Kubernetes communicate properly with the management IP address of the storage device to be connected.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-fusionstorage-scsi.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-6**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "fusionstorage-san",
                "name": "storage",
                "urls": ["https://192.168.125.20:28443"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "scsi", "portals": [{"hostname01": "192.168.125.21", "hostname02":
"192.168.125.22"}]}
            }
        ]
    }
```


Table 4-6 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the distributed storage SAN is connected, the value is fixed to **fusionstorage-san**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:28443 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.
parameters	Dictionary	Variable parameters. This parameter is mandatory.	The **protocol** parameter is fixed to **scsi**. Set **portals** to the IP address pair list of hosts and VBS nodes. The format is **[{"***hostname***":"****.*.*.**"}]**, where *hostname* indicates the host name of a worker node and **.*.*.** indicates the management IP address of a distributed storage block client. If there are multiple worker nodes, configure them in dictionary format and separate them with commas (,).

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.


```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

4.6.2 Connecting to Distributed Storage SAN over iSCSI

Perform this operation when you want to connect to distributed storage SAN over iSCSI.

Prerequisites

- An iSCSI client has been installed on all worker nodes of Kubernetes.
- All worker nodes of Kubernetes communicate properly with the management IP address of the storage device to be connected.
- All worker nodes of Kubernetes communicate properly with the service IP address of the storage device to be connected.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.
- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes.

Precautions

- The host name of a Kubernetes worker node consists of digits, letters, underscores (_), hyphens (-), periods (.), and colons (:), and must start with a digit, letter, or underscore (_). The name length cannot exceed 31 characters.
- Only FusionStorage 8.0 and later versions support iSCSI networking configuration.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-fusionstorage-iscsi.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-7**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "fusionstorage-san",
```

```
            "name": "storage",
            "urls": ["https://192.168.125.20:28443"],
            "pools": ["storagepool01", "storagepool02"],
            "parameters": {"protocol": "iscsi", "portals": ["192.168.125.21", "192.168.125.22"]}
        }
    ]
}
```

Table 4-7 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the distributed storage SAN is connected, the value is fixed to **fusionstorage-san**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:28443 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.
parameters	Dictionary	Variable parameters. This parameter is mandatory.	In scenarios where the iSCSI protocol is used, set the **protocol** parameter to a fixed value: **iscsi**. Set the **portals** parameter to the iSCSI service IP addresses of the storage device. Use commas (,) to separate multiple them. You can log in to DeviceManager to obtain them.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

4.6.3 Connecting to Distributed Storage NAS over NFS

Perform this operation when you want to connect to distributed storage NAS over NFS.

Prerequisites

- An NFS client tool has been installed on all worker nodes of Kubernetes.
- All worker nodes of Kubernetes communicate properly with the management IP address of the storage device to be connected.
- All worker nodes of Kubernetes communicate properly with the IP address of the NFS logical port on the storage device to be connected.
- You have obtained the IP address, login account, and password of any master node in the Kubernetes cluster from the administrator.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. You can also refer to the **yamls/huawei-csi-configmap-fusionstorage-nfs.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 4-8**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "fusionstorage-nas",
                "name": "storage",
                "urls": ["https://192.168.125.20:28443"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "nfs", "portals": ["192.168.125.21"]}
            }
        ]
    }
```


Table 4-8 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device. This parameter is mandatory.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage device to be connected. This parameter is mandatory.	In the scenario where the distributed storage NAS is connected, the value is fixed to **fusionstorage-nas**.
pools	List	Name of a storage pool used on the storage device to be connected. This parameter is mandatory.	One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. You can log in to DeviceManager to obtain the storage pools.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:28443 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.
parameters	Dictionary	Variable parameters in scenarios where the NAS protocol is used. This parameter is mandatory.	**portals**: logical port IP address the specified storage device. You can log in to DeviceManager to obtain it. Only one IP address can be configured.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```


Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.7 Starting huawei-csi Services

This section describes how to start huawei-csi services.

Precautions

An image may need to be downloaded during the procedure. Therefore, worker nodes in the Kubernetes cluster must be able to access external networks. In an intranet environment, obtain the image package in other ways and manually import it into all worker nodes.

Prerequisites

- You have obtained the user name and password of the storage device from the administrator.
- A local storage user whose role is administrator or higher and level is administrator is required.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Copy the **secretGenerate** tool in the Kubernetes CSI component package to any directory on the master node. For details about the tool path, see **4.2 Components in the Software Package**.

Step 3 Use an encryption tool to enter the user name and password of the storage device.

1. Run the **chmod +x secretGenerate** and **./secretGenerate** commands and enter related information as prompted.

```
# chmod +x secretGenerate
# ./secretGenerate
**************************The 1 Backend Info**************************
Current backend name is: <backend-1-name>
Current backend url is: [<backend-1-url>]
**********************************************************************
Enter backend <backend-1-name>'s user:       # Enter the user name of storage device 1.
Enter backend <backend-1-name>'s password:   # Enter the password of storage device 1.
Please enter the password again:             # Enter the password of storage device 1 again.

**********************Create CSI Secret Successful**********************
```

2. After the configuration is complete, run the **kubectl get secret -n kube-system | grep huawei-csi-secret** command to check the configuration.

```
# kubectl get secret -n kube-system | grep huawei-csi-secret
huawei-csi-secret                       Opaque                                1      8m
```

Step 4 Run the **vi huawei-csi-rbac.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the

modification is complete, press **Esc** and enter **:wq!** to save the modification. (For details, see sample file **yamls/huawei-csi-rbac.yaml** in the software package.)

NOTE

- The csi-resizer service is supported since Kubernetes v1.16.

  For details about the **huawei-csi-rbac.yaml** file, see sample file **yamls/huawei-csi-resize-rbac.yaml** in the software package.
- The csi-snapshotter service is supported since Kubernetes v1.17.

  For details about the **huawei-csi-rbac.yaml** file, see sample file **yamls/huawei-csi-resize-snapshot-rbac.yaml** in the software package.

Step 5 Run the following command to create the RBAC permission.

```
# kubectl create -f huawei-csi-rbac.yaml
```

Step 6 Run the **vi huawei-csi-controller.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. (For details, see sample file **yamls/huawei-csi-controller.yaml** in the software package.)

NOTE

- In the **huawei-csi:****.*.** configuration item in the sample .yaml file, **.*.** must be replaced with the version number of the created Huawei CSI image.
- In the **args** section of **liveness-probe** in the .yaml file, **--health-port** indicates the listening port number. To change its value, perform the following operations.
  1. In the **args** section of **liveness-probe**, modify the **health-port** parameter.

     ```
     args:
       - --csi-address=/var/lib/csi/sockets/pluginproxy/csi.sock
       - --health-port=9808
     ```

  2. In the **ports** section of **huawei-csi-driver**, modify the **containerPort** parameter.

     ```
     ports:
       - containerPort: 9808
         name: healthz
         protocol: TCP
     ```

- The csi-resizer service is supported since Kubernetes v1.16.

  For details about the **huawei-csi-controller.yaml** file, see sample file **yamls/huawei-csi-resize-controller.yaml** in the software package.
- The csi-snapshotter service is supported since Kubernetes v1.17.

  For details about the **huawei-csi-controller.yaml** file, see sample file **yamls/huawei-csi-resize-snapshot-controller.yaml** in the software package.

Step 7 Run the following command to start the controller service.

```
# kubectl create -f huawei-csi-controller.yaml
```

Step 8 Run the **vi huawei-csi-node.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. Compile the **huawei-csi-node.yaml** file. For details, see sample file **yamls/huawei-csi-node.yaml** in the software package.

NOTE

- In the **huawei-csi:****.*.** configuration item in the sample .yaml file, **.*.** must be replaced with the version number of the created Huawei CSI image.
- In the **args** section of **liveness-probe** in the .yaml file, **--health-port** indicates the listening port number. To change its value, perform the following operations.
  1. In the **args** section of **liveness-probe**, modify the **health-port** parameter.
     ```
     args:
       - --csi-address=/var/lib/csi/sockets/pluginproxy/csi.sock
       - --health-port=9800
     ```
  2. In the **ports** section of **huawei-csi-driver**, modify the **containerPort** parameter.
     ```
     ports:
       - containerPort: 9800
         name: healthz
         protocol: TCP
     ```
- In the **args** section of **huawei-csi-driver** in the .yaml file, **--volume-use-multipath** indicates that multipathing is enabled by default. The following shows how to change the value.
  ```
  args:
    - "--endpoint=/csi/csi.sock"
    - "--containerized"
    - "--driver-name=csi.huawei.com"
    - "--volume-use-multipath=false"
  ```

Step 9 Run the following command to start the node service.

```
# kubectl create -f huawei-csi-node.yaml
```

Step 10 After the huawei-csi services are deployed, run the **kubectl get pod -A | grep huawei** command to check whether the services are started.

```
# kubectl get pod -A | grep huawei
kube-system   huawei-csi-controller-695b84b4d8-tg64l   3/3   Running   0   14s
kube-system   huawei-csi-node-g6f7z                    2/2   Running   0   14s
```

----End

5 Upgrade Operations

NOTICE

- The CSI upgrade does not affect delivered resources such as PVCs, snapshots, and Pods.
- During the upgrade, CSI cannot be used to deliver new resources.

5.1 Uninstalling Original CSI

Perform this operation when you want to uninstall CSI.

Prerequisites

Before uninstalling CSI, run the **kubectl get configmap huawei-csi-configmap -n kube-system -o yaml >> huawei-csi-configmap.yaml.bak** command to back up the content of the **huawei-csi-configmap** file.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-node service (**huawei-csi-node.yaml** is the configuration information in **Step 8** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-node.yaml
```

Step 3 Run the following command to view the huawei-csi-node service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-node
```

Step 4 Run the following command to delete the huawei-csi-controller service (**huawei-csi-controller.yaml** is the configuration information in **Step 6** in **4.7 Starting huawei-csi Services**).


```
# kubectl delete -f huawei-csi-controller.yaml
```

Step 5 Run the following command to view the huawei-csi-controller service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-controller
```

Step 6 Run the following command to delete the RBAC permission (**huawei-csi-rbac.yaml** is the configuration information in **Step 4** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-rbac.yaml
```

Step 7 Run the following command to delete the huawei-csi-configmap object. By default, the **<namespace>** parameter of the huawei-csi-configmap object is **kube-system**. Replace it with the actual name.

```
# kubectl delete configmap huawei-csi-configmap -n <namespace>
```

Step 8 Run the following command to view the **configmap** information of huawei-csi. If no command output is displayed, the deletion is complete.

```
# kubectl get configmap -A | grep huawei-csi-configmap
```

Step 9 Run the following command to delete the huawei-csi-secret object. By default, the **<namespace>** parameter of the huawei-csi-secret object is **kube-system**. Replace it with the actual name.

```
# kubectl delete secret huawei-csi-secret -n <namespace>
```

Step 10 Run the following command to view the **secret** information of huawei-csi. If no command output is displayed, the deletion is complete.

```
# kubectl get secret -A | grep huawei-csi-secret
```

----End

5.2 Installing New CSI

After the uninstallation is complete, you need to reinstall the CSI.

Prerequisites

The **huawei-csi-configmap.yaml** file has been backed up.

Precautions

If the template of **huawei-csi-configmap.yaml** has changed, ensure that the following parameter settings are the same as those before the upgrade. Otherwise, huawei-csi services cannot be started and created resources cannot be managed.

- The values of **storage**, **name**, and **pools** must be the same as those in the **huawei-csi-configmap.yaml.bak** file backed up in **Prerequisites** in **5.1 Uninstalling Original CSI**.
- For details about **urls** and **parameters**, see the **huawei-csi-configmap.yaml.bak** file backed up in **Prerequisites** in **5.1 Uninstalling Original CSI** and set them based on the **huawei-csi-configmap.yaml** template of the current version. For details about the template, see **4.5 Connecting to Enterprise Storage** and **4.6 Connecting to Distributed Storage**. The following command output is only an example.


```
"backends": [
    {
        "storage": "oceanstor-san",
        "name": "***",
        "urls": ["https://*.*.*.*:8088", "https://*.*.*.*:8088"],
        "pools": ["***", "***"],
        "parameters": {"protocol": "iscsi", "portals": ["*.*.*.*", "*.*.*.*"]}
    }
]
```

Procedure

Step 1 Install CSI. For details, see **4 Installation and Deployment**.

----End

6 Instructions for Use

This chapter describes how to manage StorageClasses, PVCs, Pods, and snapshots after connecting Kubernetes to Huawei storage.

6.1 Managing a StorageClass

6.1.1 Creating a StorageClass

A StorageClass is a set of capabilities that can be selected when you apply for block storage resources. Kubernetes cluster users can create PVCs based on a StorageClass.

6.1.1.1 Creating a LUN StorageClass

This section describes how to create a LUN StorageClass.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to create a file named **StorageClass.yaml**.

```
# vi StorageClass.yaml
```

Step 3 Press **I** or **Insert** to enter the editing mode and enter the following information in the **StorageClass.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

The following shows a template of the **StorageClass.yaml** file. You can also refer to the **yamls/lun-sc-for-csi-example.yaml** example file in the software package.

Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-1**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "mysc"
provisioner: "csi.huawei.com"
parameters:
  volumeType: "lun"
  allocType: "thin"
```

Table 6-1 Parameter description

Parameter	Description	Remarks
name	User-defined name of a StorageClass object.	
provisioner	provisioner identifier.	The value is fixed to **csi.huawei.com**.
volumeType	Type of the volume to be created.	The value is fixed to **lun**.
allocType	Allocation type of the volume to be created.	This parameter is optional. The value can be **thin** or **thick**, and the default value is **thin**.
cloneSpeed	Clone speed.	This parameter is optional. The value ranges from 1 to 4 and the default value is **3**. **4** indicates the highest speed. This parameter is available when you clone a PVC or create a PVC using a snapshot. For details, see **6.2.3 (Optional) Cloning a PVC** or **6.2.4 (Optional) Creating a PVC Using a Snapshot**.
fsType	File system type.	This parameter is optional. The value can be **ext2**, **ext3**, or **ext4**, and the default value is **ext4**.

Step 4 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorageClass.yaml
```

Step 5 Run the following command to view the information about the created StorageClass.

```
# kubectl get sc
NAME   PROVISIONER      AGE
mysc   csi.huawei.com   1h
```

----End

6.1.1.2 Creating a File System StorageClass

This section describes how to create a file system StorageClass.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to create a file named **StorageClass.yaml**.

```
# vi StorageClass.yaml
```

Step 3 Press **I** or **Insert** to enter the editing mode and enter the following information in the **StorageClass.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

The following shows a template of the **StorageClass.yaml** file. You can also refer to the **yamls/fs-sc-for-csi-example.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-2**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "mysc"
provisioner: "csi.huawei.com"
parameters:
  volumeType: "fs"
  allocType: "thin"
  authClient: "*"
```

Table 6-2 Parameter description

Parameter	Description	Remarks
name	User-defined name of a StorageClass object.	
provisioner	provisioner identifier.	The value is fixed to **csi.huawei.com**.
volumeType	Type of the volume to be created.	The value is fixed to **fs**.
authClient	Client that can access the volume.	This parameter is mandatory. You can enter the client host name, client IP address, or client IP address segment, or use an asterisk (*) to represent all client IP addresses. You can specify multiple clients which are separated by semicolons (;).
allocType	Allocation type of the volume to be created.	This parameter is optional. The value can be **thin** or **thick**, and the default value is **thin**.

Parameter	Description	Remarks
cloneSpeed	Clone speed.	This parameter is optional. The value ranges from 1 to 4 and the default value is **3**. **4** indicates the highest speed. This parameter is available when you clone a PVC or create a PVC using a snapshot. For details, see **6.2.3 (Optional) Cloning a PVC** or **6.2.4 (Optional) Creating a PVC Using a Snapshot**.

Step 4 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorageClass.yaml
```

Step 5 Run the following command to view the information about the created StorageClass.

```
# kubectl get sc
NAME   PROVISIONER      AGE
mysc   csi.huawei.com   1h
```

----End

6.1.2 Deleting a StorageClass

This section describes how to delete a StorageClass.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to query StorageClasses in the cluster.

```
# kubectl get sc
NAME         PROVISIONER      AGE
huawei-nas   csi.huawei.com   3s
mysc         csi.huawei.com   16s
```

Step 3 Run the following command to delete a StorageClass. For example, delete the StorageClass named *mysc*.

```
# kubectl delete sc mysc
storageclass.storage.k8s.io "mysc" deleted
```

Step 4 Run the following command to query StorageClasses in the cluster. If the command output does not contain the name of the StorageClass you want to delete, it is successfully deleted.

```
# kubectl get sc
NAME         PROVISIONER      AGE
huawei-nas   csi.huawei.com   11s
```

----End

6.2 Managing a PVC

6.2.1 Creating a PVC

Perform this operation when you want to create a PVC.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi PersistentVolumeClaim.yaml** command to create a file named **PersistentVolumeClaim.yaml**.

```
# vi PersistentVolumeClaim.yaml
```

Step 3 Press **I** or **Insert** to enter the editing mode and enter the following information in the **PersistentVolumeClaim.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

The following shows a template of the **PersistentVolumeClaim.yaml** file. You can also refer to the **yamls/pvc-example.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-3**.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: "mypvc"
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: "mysc"
  resources:
    requests:
      storage: 100Gi
```

Table 6-3 Parameter description

Parameter	Description	Remarks
name	User-defined name of a PVC object.	
storageClassName	Name of the StorageClass object.	Enter the name of the StorageClass object created in **6.1 Managing a StorageClass**.

Parameter	Description	Remarks
storage	Size of the volume to be created.	The value format is ***Gi. The unit is GiB. • The PVC capacity depends on storage specifications and host specifications. The following uses the connection between OceanStor Dorado V6 6.1.2/ OceanStor Pacific series 8.1.0 and CentOS 7 as an example. See Table 6-4. • For other storage devices and hosts, check the specifications according to the value of **VolumeType** in StorageClass. – If the value of **volumeType** is **lun**, refer to the storage specifications. For details, see **https://info.support.huawei.com/storage/spec/#/home**. In addition, refer to the host connectivity guide at **https://support.huawei.com/enterprise/en/doc/EDOC1100113070/e067543b**. – If the value of **volumeType** is **fs**, refer to the storage specifications. For details, see **https://info.support.huawei.com/storage/spec/#/home**. • If the PVC capacity does not meet the specifications, a PVC or Pod may fail to be created due to the limitations of storage specifications or host file system specifications.

Parameter	Description	Remarks
accessModes	Access mode of the volume.	LUN volumes support **ReadWriteOnce**, **ReadOnlyMany**, and **ReadWriteMany**. If the **ReadWriteMany** mode is used and multiple Pods access the volume at the same time, the Pod service must ensure data consistency. File system volumes support **ReadWriteOnce**, **ReadOnlyMany**, and **ReadWriteMany**.

Table 6-4 PVC capacity specifications

volumeType	Storage Type	Storage Specifications	ext4 Specifications	CSI Specifications
lun	OceanStor Dorado V6 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti
	OceanStor Pacific series 8.1.0	64 Mi to 512 Ti	50 Ti	64 Mi to 50 Ti
fs	OceanStor Dorado V6 6.1.2	1 Gi to 32 Pi	N/A	1 Gi to 32 Pi
	OceanStor Pacific series 8.1.0	1 Ki to 256 Pi	N/A	1 Ki to 256 Pi

Step 4 Run the following command to create a PVC based on the .yaml file.

```
# kubectl create -f PersistentVolumeClaim.yaml
```

Step 5 After a period of time, run the following command to view the information about the created PVC.

```
# kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mypvc     Bound    pvc-840054d3-1d5b-4153-b73f-826f980abf9e   100Gi      RWX            mysc           12s
```

NOTE

After the PVC is created, if the PVC is in the **Pending** state, see **9.6 When a PVC Is Created, the PVC Is in the Pending State**.

----End

6.2.2 (Optional) Expanding the Capacity of a PVC

6.2.2.1 Installing the Expansion-Dependent Component Service

Perform the operations instructed in **4.7 Starting huawei-csi Services**. Use **huawei-csi-resize-snapshot-controller.yaml** to deploy the huawei-csi services and run the following command to check whether the huawei-csi services are started.

```
# kubectl get pod -A | grep huawei
kube-system     huawei-csi-controller-fd5f97768-qlldc     6/6     Running          0          16s
kube-system     huawei-csi-node-25txd                     2/2     Running          0          15s
```

6.2.2.2 Expanding the Capacity of a PVC

This section describes how to expand the capacity of a PVC.

Prerequisites

A PVC has been created. For details, see **6.2.1 Creating a PVC**.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to edit the new **StorageClass.yaml** file. Add the **allowVolumeExpansion** configuration item to the file. The following is an example:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
allowVolumeExpansion: true
```

Step 3 Create a StorageClass. For details, see **6.1 Managing a StorageClass**. If a StorageClass with the same name already exists, specify **name** in the **StorageClass.yaml** file again.

Step 4 Create a PVC. For details, see **6.2.1 Creating a PVC**. If a PVC with the same name already exists, specify **name** in the **PersistentVolumeClaim.yaml** file again.

Step 5 Run the following command to expand the capacity.

```
# kubectl patch pvc mypvc -p '{"spec":{"resources":{"requests":{"storage":"120Gi"}}}}'
```

In the preceding command, *mypvc* indicates the name of the PVC to be expanded, and *120Gi* indicates the capacity after expansion. Change the values based on the site requirements.

NOTE

- The PVC capacity depends on storage specifications and host specifications. The following uses the connection between OceanStor Dorado V6 6.1.2/OceanStor Pacific series 8.1.0 and CentOS 7 as an example. See **Table 6-5**.
- For other storage devices and hosts, check the specifications according to the value of **VolumeType** in **StorageClass**.
  - If the value of **volumeType** is **lun**, refer to the storage specifications. For details, see **https://info.support.huawei.com/storage/spec/#/home**. In addition, refer to the host connectivity guide at **https://support.huawei.com/enterprise/en/doc/EDOC1100113070/e067543b**.
  - If the value of **volumeType** is **fs**, refer to the storage specifications. For details, see **https://info.support.huawei.com/storage/spec/#/home**.
- If the PVC capacity does not meet the specifications, a PVC or Pod may fail to be created due to the limitations of storage specifications or host file system specifications.

Table 6-5 PVC capacity specifications

volumeType	Storage Type	Storage Specifications	ext4 Specifications	CSI Specifications
lun	OceanStor Dorado V6 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti
	OceanStor Pacific series 8.1.0	64 Mi to 512 Ti	50 Ti	64 Mi to 50 Ti
fs	OceanStor Dorado V6 6.1.2	1 Gi to 32 Pi	N/A	1 Gi to 32 Pi
	OceanStor Pacific series 8.1.0	1 Ki to 256 Pi	N/A	1 Ki to 256 Pi

Step 6 Run the following command to check whether the capacity changes.

```
# kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mypvc     Bound    pvc-840054d3-1d5b-4153-b73f-826f980abf9e   120Gi      RWX            mysc           24s
```

----End

6.2.3 (Optional) Cloning a PVC

Perform this operation when you want to clone an existing PVC on Kubernetes.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **PersistentVolumeClaim.yaml** file. The following shows a template of the **PersistentVolumeClaim.yaml** file. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-6**.

```
kind: PersistentVolumeClaim
apiVersion: v1
```



```
metadata:
  name: ***
spec:
  storageClassName: ***
  dataSource:
    name: ***
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: ***Gi
```

Table 6-6 Parameter description

Parameter	Description	Remarks
metadata/ name	User-defined name of a new PVC object.	
spec/ storageClassName	Name of the StorageClass object.	Enter the name of the StorageClass object created in **6.1 Managing a StorageClass**. The value must be the same as the name of the StorageClass in **dataSource**.
spec/ dataSource/ name	Name of the source PVC object.	
spec/ resources/ requests/ storage	Size of the volume to be created.	The value must be greater than or equal to the size of the source PVC. The value format is ***Gi**. The unit is GiB.

Step 3 Run the **vi PersistentVolumeClaim.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Step 4 Run the following command to create a PVC based on the .yaml file.

```
# kubectl create -f PersistentVolumeClaim.yaml
```

----End

6.2.4 (Optional) Creating a PVC Using a Snapshot

Perform this operation when you want to create a PVC for an existing snapshot on Kubernetes.

Prerequisites

A snapshot has been created. For details, see **6.4 (Optional) Managing a Snapshot**.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **PersistentVolumeClaim.yaml** file. The following shows a template of the **PersistentVolumeClaim.yaml** file. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-7**.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ***
spec:
  storageClassName: ***
  dataSource:
    name: ***
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: ***Gi
```

Table 6-7 Parameter description

Parameter	Description	Remarks
metadata/ name	User-defined name of a new PVC object.	
spec/ storageClassName	Name of the StorageClass object.	Enter the name of the StorageClass object created in **6.1 Managing a StorageClass**. The value must be the same as the name of the StorageClass of the original PVC in **dataSource**.
spec/ dataSource/ name	Name of the source VolumeSnapshot object.	
spec/ resources/ requests/ storage	Size of the volume to be created.	The value must be greater than or equal to the size of the source VolumeSnapshot. The value format is ***Gi**. The unit is GiB.

Step 3 Run the **vi PersistentVolumeClaim.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Step 4 Run the following command to create a PVC based on the .yaml file.

```
# kubectl create -f PersistentVolumeClaim.yaml
```

----End

6.2.5 Deleting a PVC

This section describes how to delete a PVC.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to query PVCs in the cluster.

```
# kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mypvc     Bound    pvc-840054d3-1d5b-4153-b73f-826f980abf9e   100Gi      RWX            mysc           12s
```

NOTE

Before deleting a PVC, if the PVC is in the **Pending** state, you are not advised to directly delete the PVC. To delete the PVC, see **9.7 Before a PVC Is Deleted, the PVC Is in the Pending State**.

Step 3 Run the following command to delete a PVC. For example, delete the PVC named *mypvc*.

```
# kubectl delete pvc mypvc
persistentvolumeclaim "mypvc" deleted
```

Step 4 Run the following command to query PVCs in the cluster. If the command output does not contain the name of the PVC you want to delete, it is successfully deleted.

```
# kubectl get pvc
No resources found in default namespace.
```

----End

6.3 Managing a Pod

6.3.1 Creating a Pod

A Pod is an original storage pool or storage function set. It function as the container of virtual volumes, which means only the storage container allocates storage space to virtual volumes. This operation enables you to quickly obtain specified storage resources.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi Pod.yaml** command to create a file named **Pod.yaml**.

```
# vi Pod.yaml
```

Step 3 Press **I** or **Insert** to enter the editing mode and enter the following information in the **Pod.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

The following shows a template of the **Pod.yaml** file. You can also refer to the **yamls/pod-example.yaml** example file in the software package. Set related

parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-8**.

```
kind: Pod
apiVersion: v1
metadata:
  name: "mypod"
spec:
  containers:
    - name: "mycontainer"
      image:  "***"
      volumeMounts:
        - name: mypv
          mountPath: "/mnt/path/in/container"
  volumes:
    - name: mypv
      persistentVolumeClaim:
        claimName: "mypvc"
```

Table 6-8 Parameter description

Parameter	Description	Remarks
metadata:name	User-defined name of a Pod object.	
spec:containers:name	User-defined container name.	
spec:containers:image	Container image.	Set this parameter based on the site requirements.
spec:containers:image:volumeMounts:mountPath	Volume mount path in the container.	
spec:volumes:persistentVolumeClaim:claimName	Name of the PVC object.	Enter the name of the PVC object created in **6.2.1 Creating a PVC**.

Step 4 Run the following command to create a Pod based on the .yaml file.

```
# kubectl create -f Pod.yaml
```

Step 5 Run the following command to view the information about the created Pod.

```
# kubectl get pod
NAME       READY   STATUS    RESTARTS   AGE
mypod      1/1     Running   0          37s
```

----End

6.3.2 Deleting a Pod

This section describes how to delete a Pod.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to query Pods in the cluster.

```
# kubectl get pod
NAME     READY   STATUS   RESTARTS   AGE
mypod    1/1     Running  0          14h
```

Step 3 Run the following command to delete a Pod. For example, delete the Pod named *mypod*.

```
# kubectl delete pod mypod
pod "mypod" deleted
```

Step 4 Run the following command to query Pods in the cluster. If the command output does not contain the name of the Pod you want to delete, it is successfully deleted.

```
# kubectl get pod
No resources found in default namespace.
```

----End

6.4 (Optional) Managing a Snapshot

CSI supports snapshot v1beta1 since Kubernetes v1.17. For details, see **https://kubernetes-csi.github.io/docs/external-snapshotter.html**.

6.4.1 Installing the Snapshot-Dependent Component Service

Perform the following steps only on any master node.

Prerequisites

Before the installation, run the **kubectl api-resources | grep snapshot | awk '{print $1}'** command on the master node to check whether the snapshot-related resource service has been installed. If the following information is displayed, you do not need to install it again.

```
# kubectl api-resources | grep snapshot | awk '{print $1}'
volumesnapshotclasses
volumesnapshotcontents
volumesnapshots
```

If the command output does not contain the preceding service, perform the following operations to install the service.

Step 1 Compile the **volumesnapshotclasses.yaml** file if the **volumesnapshotclasses** resource is missing. You can refer to the **yamls/snapshot.storage.k8s.io_volumesnapshotclasses.yaml** example file in the software package.

```
# kubectl create -f volumesnapshotclasses.yaml
```

Step 2 Compile the **volumesnapshotcontents.yaml** file if the **volumesnapshotcontents** resource is missing. You can refer to the **yamls/snapshot.storage.k8s.io_volumesnapshotcontents.yaml** example file in the software package.

```
# kubectl create -f volumesnapshotcontents.yaml
```

Step 3 Compile the **volumesnapshots.yaml** file if the **volumesnapshots** resource is missing. You can refer to the **yamls/snapshot.storage.k8s.io_volumesnapshots.yaml** example file in the software package.


```
# kubectl create -f volumesnapshots.yaml
```

Step 4 If huawei-csi services have been started, perform the operations instructed in **8.1 Uninstalling CSI**.

Step 5 Perform the operations instructed in **4.7 Starting huawei-csi Services**. Use **huawei-csi-resize-snapshot-controller.yaml** to deploy the huawei-csi services and run the following command to check whether the huawei-csi services are started.

```
# kubectl get pod -A | grep huawei
kube-system     huawei-csi-controller-fd5f97768-qlldc     6/6     Running         0          16s
kube-system     huawei-csi-node-25txd                     2/2     Running         0          15s
```

----End

6.4.2 Managing a VolumeSnapshotClass

6.4.2.1 Creating a VolumeSnapshotClass

This section describes how to create a VolumeSnapshotClass.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi VolumeSnapshotClass.yaml** command to create a file named **VolumeSnapshotClass.yaml**.

```
# vi VolumeSnapshotClass.yaml
```

Step 3 Press **I** or **Insert** to enter the editing mode and enter the following information in the **VolumeSnapshotClass.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

The following shows a template of the **VolumeSnapshotClass.yaml** file. You can also refer to the **yamls/snapshotclass.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-9**.

```
apiVersion: snapshot.storage.k8s.io/v1beta1
kind: VolumeSnapshotClass
metadata:
  name: mysnapclass
driver: csi.huawei.com
deletionPolicy: Delete
```

Table 6-9 Parameter description

Parameter	Description	Remarks
name	User-defined name of a VolumeSnapshotClass object.	
driver	driver identifier.	The value is fixed to **csi.huawei.com**.

Parameter	Description	Remarks
deletionPolicy	Handles the **VolumeSnapshotContent** policy when a VolumeSnapshot is deleted.	This parameter is mandatory. The value can be **Delete** or **Retain**.

Step 4 Run the following command to create a VolumeSnapshotClass based on the .yaml file.

```
# kubectl create -f VolumeSnapshotClass.yaml
```

Step 5 Run the following command to view the information about the created VolumeSnapshotClass.

```
# kubectl get volumesnapshotclass
NAME          DRIVER          DELETIONPOLICY   AGE
mysnapclass   csi.huawei.com   Delete          25s
```

----End

6.4.2.2 Deleting a VolumeSnapshotClass

This section describes how to delete a VolumeSnapshotClass.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to query VolumeSnapshotClasses in the cluster.

```
# kubectl get volumesnapshotclass
NAME          DRIVER          DELETIONPOLICY   AGE
mysnapclass   csi.huawei.com   Delete          52s
```

Step 3 Run the following command to delete a VolumeSnapshotClass. For example, delete the VolumeSnapshotClass named *mysnapclass*.

```
# kubectl delete volumesnapshotclass mysnapclass
volumesnapshotclass.snapshot.storage.k8s.io "mysnapclass" deleted
```

Step 4 Run the following command to query VolumeSnapshotClasses in the cluster. If the command output does not contain the name of the VolumeSnapshotClass you want to delete, it is successfully deleted.

```
# kubectl get volumesnapshotclass
No resources found in default namespace.
```

----End

6.4.3 Managing a VolumeSnapshot

6.4.3.1 Creating a VolumeSnapshot

This section describes how to create a VolumeSnapshot.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi VolumeSnapshot.yaml** command to create a file named **VolumeSnapshot.yaml**.

```
# vi VolumeSnapshot.yaml
```

Step 3 Press **I** or **Insert** to enter the editing mode and enter the following information in the **VolumeSnapshot.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

The following shows a template of the **VolumeSnapshot.yaml** file. You can also refer to the **yamls/snapshot.yaml** example file in the software package. Set related parameters based on the site requirements and save the file in yaml format. For details, see **Table 6-10**.

```
apiVersion: snapshot.storage.k8s.io/v1beta1
kind: VolumeSnapshot
metadata:
  name: mysnapshot
spec:
  volumeSnapshotClassName: mysnapclass
  source:
    persistentVolumeClaimName: mypvc
```

Table 6-10 Parameter description

Parameter	Description	Remarks
name	User-defined name of a VolumeSnapshot object.	
volumeSnapshotClassName	Name of the VolumeSnapshotClass object.	Enter the name of the VolumeSnapshotClass object created in **6.4.2.1 Creating a VolumeSnapshotClass**.
persistentVolumeClaimName	Name of the source PVC object.	Enter the name of the PVC object created in **6.2.1 Creating a PVC**.

Step 4 Run the following command to create a VolumeSnapshot based on the .yaml file.

```
# kubectl create -f VolumeSnapshot.yaml
```

Step 5 Run the following command to view the information about the created VolumeSnapshot.

```
# kubectl get volumesnapshot
NAME         READYTOUSE   SOURCEPVC   SOURCESNAPSHOTCONTENT   RESTORESIZE
SNAPSHOTCLASS   SNAPSHOTCONTENT                                   CREATIONTIME   AGE
mysnapshot   true         mypvc                                   100Gi         mysnapclass
snapcontent-1009af0a-24c2-4435-861c-516224503f2d   <invalid>     78s
```

----End

6.4.3.2 Deleting a VolumeSnapshot

This section describes how to delete a VolumeSnapshot.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to query VolumeSnapshots in the cluster.

```
# kubectl get volumesnapshot
NAME         READYTOUSE   SOURCEPVC   SOURCESNAPSHOTCONTENT   RESTORESIZE
SNAPSHOTCLASS   SNAPSHOTCONTENT                                   CREATIONTIME   AGE
mysnapshot   true         mypvc                                   100Gi         mysnapclass
snapcontent-1009af0a-24c2-4435-861c-516224503f2d   <invalid>      78s
```

Step 3 Run the following command to delete a VolumeSnapshot. For example, delete the VolumeSnapshot named *mysnapshot*.

```
# kubectl delete volumesnapshot mysnapshot
volumesnapshot.snapshot.storage.k8s.io "mysnapshot" deleted
```

Step 4 Run the following command to query VolumeSnapshots in the cluster. If the command output does not contain the name of the VolumeSnapshot you want to delete, it is successfully deleted.

```
# kubectl get volumesnapshot
No resources found in default namespace.
```

----End

7 Advanced Features

This chapter describes how to configure advanced features of Huawei storage.

7.1 Configuring Multiple Backends

Huawei CSI supports multiple backends. Perform this operation when you want to configure multiple backends.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure the **huawei-csi-configmap.yaml** file. The following shows a template of the **huawei-csi-configmap.yaml** file. Set related parameters based on the site requirements and save the file in yaml format.

Multiple backends are separated by commas (,). For details about each backend, see **4.5 Connecting to Enterprise Storage** or **4.6 Connecting to Distributed Storage**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
```



```
    "backends": [
      {
        "storage": "***",
        "name": "backend1",
        ...
      },
      {
        "storage": "***",
        "name": "backend2",
        ...
      }
    ]
}
```

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

----End

7.2 Creating a PVC for a Specified Backend

When multiple backends are configured, you can perform the following operations to create a PVC for a specified backend.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

- Add the **backend** configuration item under **parameters**.
- The value of **backend** is the name of a backend in **huawei-csi-configmap.yaml**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  backend: "***"
```

Step 3 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorageClass.yaml
```

----End

7.3 Creating a PVC for a Specified Storage Pool

When multiple storage pools are configured, you can perform the following operations to create a PVC for a specified storage pool.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

- Add the **backend** configuration item under **parameters**.
- The value of **pool** is the name of a storage pool in **huawei-csi-configmap.yaml**.

NOTE

The volume to be created using the StorageClass will be created in the specified storage pool. The existing PVC will not change the storage pool information.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  pool: "***"
```

Step 3 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorageClass.yaml
```

----End

7.4 Configuring ALUA

7.4.1 Configuring ALUA for OceanStor V3/V5 and OceanStor Dorado V3

This section describes how to configure ALUA if multipathing is used during the connection to block storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add ALUA parameters under the **parameters** section. For details, see **Table 7-1**.

```
{
  "backends": [
  {
    "storage": "oceanstor-san",
    ...
    "parameters": {..., "ALUA": {"<HostName>": {"MULTIPATHTYPE": "*", "FAILOVERMODE": "*",
"SPECIALMODETYPE": "*", "PATHTYPE": "*"}, "<HostName>": {...}}}
  }
  ]
}
```

Table 7-1 ALUA parameter description

Parameter	Description	Remarks
<HostName>	The value of **HostName** is the host name of a worker node.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. For details about the configuration rules and priorities, see the following note.
MULTIPATHTYPE	Multipathing type. The value can be: ● **0**: default ● **1**: third-party multipathing	
FAILOVERMODE	Initiator switchover mode. The value can be: ● **0**: early-version ALUA ● **1**: common ALUA ● **2**: ALUA not used ● **3**: special ALUA	This parameter needs to be specified only when third-party multipathing is used. All OceanStor V5 models do not support early-version ALUA.
SPECIALMODETYPE	Special mode type of the initiator. The value can be: ● **0**: special mode 0 ● **1**: special mode 1 ● **2**: special mode 2 ● **3**: special mode 3	This parameter needs to be specified only when the initiator switchover mode is special ALUA.
PATHTYPE	Initiator path type. The value can be: ● **0**: preferred path ● **1**: non-preferred path	This parameter needs to be specified only when third-party multipathing is used.

NOTE

- The ALUA configuration may vary according to the OS. Visit **https://support.huawei.com/enterprise/en/index.html**, enter **Host Connectivity Guide** in the search box, and click the search button. In the search result, select the host connectivity guide for the desired OS and configure ALUA based on the recommended configurations in the guide.
- A node with a Pod provisioned does not proactively change ALUA information. The host ALUA configuration changes only after a Pod is provisioned again to the node.
- The value of **HostName** is a regular expression. For details about how to configure it, see **Regular expression**.

  When **HostName** is set to *****, the common configuration is used and takes effect on hosts with any name. When **HostName** is set to another value, the general configuration is used. When you configure **HostName**, the number of host connections is limited. For details about the limitation, see **Specifications Query** and search for **Maximum number of iSCSI connections per controller enclosure**. If the number of host connections is less than or equal to the specifications, you are advised to use the general configuration. If the number of host connections is greater than the specifications, you are advised to use the common configuration.

  Configuration policy rules:

  – Priority: General host name configuration > Common host name configuration. For details, see example 1 in **10.1 Example ALUA Configuration Policy of OceanStor V3/V5 and OceanStor Dorado V3**.

  – In the general configuration, use the first ALUA section that meets the configuration policy. For details, see example 2 in **10.1 Example ALUA Configuration Policy of OceanStor V3/V5 and OceanStor Dorado V3**.

  – In the general configuration, if you need to exactly match a host, refer to example 3 in **10.1 Example ALUA Configuration Policy of OceanStor V3/V5 and OceanStor Dorado V3**.
- OceanStor V3/V5 and OceanStor Dorado V3 use this configuration mode. For details about related parameters, see **Table 7-2**.

Table 7-2 Recommended ALUA parameter configurations for OceanStor V3/V5 and OceanStor Dorado V3

Scenario	Host Type	Whether the Storage Has the Preferred Path	Recommended ALUA Configuration
HyperMetro storage	CentOS/RHEL host	Storage with the preferred path	ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"
		Storage with the non-preferred path	ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="1"

Scenario	Host Type	Whether the Storage Has the Preferred Path	Recommended ALUA Configuration
	Ubuntu/SUSE/Debian host	Storage with the preferred path	ALUA="1" FAILOVERMODE="1" PATHTYPE="0"
		Storage with the non-preferred path	ALUA="1" FAILOVERMODE="1" PATHTYPE="1"
Non-HyperMetro storage	CentOS/RHEL host	N/A	ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"
	Ubuntu/SUSE/Debian host	N/A	ALUA="1" FAILOVERMODE="1" PATHTYPE="0"

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

----End

7.4.2 Configuring ALUA for OceanStor Dorado V6

This section describes how to configure ALUA if multipathing is used during the connection to block storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add ALUA parameters under the **parameters** section. For details, see **Table 7-3**.


```
{
  "backends": [
  {
    "storage": "oceanstor-san",
    ...
    "parameters": {..., "ALUA": {"<HostName>": {"accessMode": "*", "hyperMetroPathOptimized": "*"},
"<HostName>": {...}}}
  }
  ]
}
```

Table 7-3 ALUA parameter description

Parameter	Description	Remarks
<HostName>	The value of **HostName** is the host name of a worker node.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. For details about the configuration rules and priorities, see the following note.
accessMode	Host access mode. The value can be: ● **0**: balanced mode ● **1**: asymmetric mode	--
hyperMetroPathOptimized	Whether the path of the host on the current storage array is preferred in HyperMetro scenarios. The value can be: ● **1**: yes ● **0**: no	This parameter needs to be specified only when the host access mode is set to asymmetric.

NOTE

- The ALUA configuration may vary according to the OS. Visit **https://support.huawei.com/enterprise/en/index.html**, enter **Host Connectivity Guide** in the search box, and click the search button. In the search result, select the host connectivity guide for the desired OS and configure ALUA based on the recommended configurations in the guide.
- A node with a Pod provisioned does not proactively change ALUA information. The host ALUA configuration changes only after a Pod is provisioned again to the node.
- The value of **HostName** is a regular expression. For details about how to configure it, see **Regular expression**.

  When **HostName** is set to *****, the common configuration is used and takes effect on hosts with any name. When **HostName** is set to another value, the general configuration is used. When you configure **HostName**, the number of host connections is limited. For details about the limitation, see **Specifications Query** and search for **Maximum number of iSCSI connections per controller enclosure**. If the number of host connections is less than or equal to the specifications, you are advised to use the general configuration. If the number of host connections is greater than the specifications, you are advised to use the common configuration.

  Configuration policy rules:

  – Priority: General host name configuration > Common host name configuration. For details, see example 1 in **10.2 Example ALUA Configuration Policy of OceanStor Dorado V6**.

  – In the general configuration, use the first ALUA section that meets the configuration policy. For details, see example 2 in **10.2 Example ALUA Configuration Policy of OceanStor Dorado V6**.

  – In the general configuration, if you need to exactly match a host, refer to example 3 in **10.2 Example ALUA Configuration Policy of OceanStor Dorado V6**.
- If a host uses only OceanStor Dorado V6 all-flash storage, see **Table 7-4** for detailed parameters.
- If you add OceanStor Dorado V6 all-flash storage to a host that uses OceanStor converged storage, see **Table 7-5** for detailed parameters.

Table 7-4 Recommended ALUA parameter configurations for OceanStor Dorado V6 all-flash storage

Scenario	Host Type	Host Access Mode	Recommended ALUA Configuration
HyperMetro storage	CentOS/RHEL/Ubuntu/SUSE/Debian host	Load balancing mode	ALUA not required
		Asymmetric mode + Storage with the preferred path	ACCESSMODE="1" HYPERMETROPATHOPTIMIZED="1"
		Asymmetric mode + Storage with the non-preferred path	ACCESSMODE="1" HYPERMETROPATHOPTIMIZED="0"

Scenario	Host Type	Host Access Mode	Recommended ALUA Configuration
Non-HyperMetro storage	CentOS/RHEL/Ubuntu/SUSE/Debian host	N/A	ALUA not required

Table 7-5 Recommended ALUA parameter configurations for hybrid OceanStor V3/V5, OceanStor Dorado V3, and OceanStor Dorado V6

Scenario	Host Type	Host Access Mode	Recommended ALUA Configuration
HyperMetro storage	CentOS/RHEL/Ubuntu/SUSE/Debian host	Load balancing mode	ACCESSMODE="1" HYPERMETROPATHOPTIMIZED="1"
		Asymmetric mode + Storage with the preferred path	ACCESSMODE="1" HYPERMETROPATHOPTIMIZED="1"
		Asymmetric mode + Storage with the non-preferred path	ACCESSMODE="1" HYPERMETROPATHOPTIMIZED="0"
Non-HyperMetro storage	CentOS/RHEL/Ubuntu/SUSE/Debian host	N/A	ACCESSMODE="1" HYPERMETROPATHOPTIMIZED="1"

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

----End

7.4.3 Configuring ALUA for Distributed Storage

This section describes how to configure ALUA if multipathing is used during the connection to block storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add ALUA parameters under the **parameters** section. For details, see **Table 7-6**.

```
{
  "backends": [
  {
    "storage": "fusionstorage-san",
    ...
    "parameters": {..., "ALUA": {"<HostName>": {"switchoverMode": "*", "pathType": "*"},
"<HostName>": {...}}}
  }
  ]
}
```

Table 7-6 ALUA parameter description

Parameter	Description	Remarks
<HostName>	The value of **HostName** is the host name of a worker node.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. For details about the configuration rules and priorities, see the following note.
switchoverMode	Switchover mode. The value can be: • **Disable_alua**: disables ALUA. • **Enable_alua**: enables ALUA.	--
pathType	Path type. The value can be: • **optimal_path**: preferred path • **non_optimal_path**: non-preferred path	--

NOTE

- Only the iSCSI scenario of distributed storage is supported.
- A node with a Pod provisioned does not proactively change ALUA information. The host ALUA configuration changes only after a Pod is provisioned again to the node.
- The value of **HostName** is a regular expression. For details about how to configure it, see **Regular expression**.

  When **HostName** is set to *, the common configuration is used and takes effect on hosts with any name. When **HostName** is set to another value, the general configuration is used.

  Configuration policy rules:

  – Priority: General host name configuration > Common host name configuration. For details, see example 1 in **10.3 Example ALUA Configuration Policy of Distributed Storage**.

  – In the general configuration, use the first ALUA section that meets the configuration policy. For details, see example 2 in **10.3 Example ALUA Configuration Policy of Distributed Storage**.

  – In the general configuration, if you need to exactly match a host, refer to example 3 in **10.3 Example ALUA Configuration Policy of Distributed Storage**.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

----End

7.5 Configuring Storage Topology Awareness

In the Kubernetes cluster, resources can be scheduled and provisioned based on the topology labels of nodes and the topology capabilities supported by storage backends.

Prerequisites

- Kubernetes v1.17 and later versions support the topology awareness feature.
- You need to configure topology labels on worker nodes in the cluster. The method is as follows:

  a. Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

  b. Run the **kubectl get node** command to view information about worker nodes in the current cluster.

```
# kubectl get node
NAME     STATUS   ROLES                  AGE   VERSION
node01   Ready    controlplane,etcd,worker   42d   v1.19.3
node02   Ready    worker                 42d   v1.19.3
node03   Ready    worker                 42d   v1.19.3
```


c. Run the **kubectl label node** *<nodename>* **topology.kubernetes.io/***<key>*=*<value>* command to configure a topology label for a worker node. In the preceding command, *<nodename>* indicates the name of a worker node. For details about the **key** and **value** parameters, see **Table 7-7**.

```
# kubectl label node node01 topology.kubernetes.io/zone=ChengDu
node/node01 labeled
```

Table 7-7 Parameter description

Parameter	Description	Remarks
<key>	Unique identifier of a topology label.	The value can be **zone**, **region**, or **protocol.***<protocol>*. *<protocol>* can be set to **iscsi**, **nfs**, **fc**, or **roce**.
<value>	Value of a topology label.	If **key** is set to **zone** or **region**, **value** is a user-defined parameter. If **key** is set to **protocol.***<protocol>*, **value** is fixed to **csi.huawei.com**.

NOTE

- A topology label must start with **topology.kubernetes.io**. Topology label examples:
  - Example 1: **topology.kubernetes.io/region=China-west**
  - Example 2: **topology.kubernetes.io/zone=ChengDu**
  - Example 3: **topology.kubernetes.io/protocol.iscsi=csi.huawei.com**
  - Example 4: **topology.kubernetes.io/protocol.fc=csi.huawei.com**
- A key in a topology label on a node can have only one value.
- If multiple protocols are configured in a topology label on a node, when you select a backend, the backend needs to meet only one of the protocols.
- If both the region and the zone are configured in a topology label on a node, when you select a backend, the backend must meet both of them.

d. Run the **kubectl get nodes -o=jsonpath='{range .items[*]}[{.metadata.name}, {.metadata.labels}]{"\n"}{end}' | grep --color "topology.kubernetes.io"** command to view the label information about all worker nodes in the current cluster.

```
# kubectl get nodes -o=jsonpath='{range .items[*]}[{.metadata.name}, {.metadata.labels}]{"\n"}
{end}' | grep --color "topology.kubernetes.io"
[node01, {"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/
arch":"amd64","kubernetes.io/hostname":"node01","kubernetes.io/os":"linux","node-
role.kubernetes.io/controlplane":"true","node-role.kubernetes.io/etcd":"true","node-
role.kubernetes.io/worker":"true","topology.kubernetes.io/zone":"ChengDu"}]
```


Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add the **supportedTopologies** field under the **backends** section in the **huawei-csi-configmap.yaml** file to configure the topology information supported by each backend. The following is a backend example.

```
{
  "backends":[
  {
    "storage": "oceanstor-san",
    ...
    "parameters": {"protocol": "iscsi", "portals": ["192.168.125.22", "192.168.125.23"]},
    "supportedTopologies": [
      {"topology.kubernetes.io/region": "China-west", "topology.kubernetes.io/zone": "ChengDu"},
      {"topology.kubernetes.io/region": "China-south", "topology.kubernetes.io/zone": "ShenZhen"}]
  }
  ]
}
```

NOTE

- **supportedTopologies** is a list. Each element in the list is a dictionary.
- Only **topology.kubernetes.io/region** or **topology.kubernetes.io/zone** can be configured for each element in the list. The parameter value must be the same as the **topology label set in the prerequisites**. (**topology.kubernetes.io/protocol.<protocol>** cannot be configured.)

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

Step 6 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and add related parameters in the .yaml file. For details about the parameters, see **Table 7-8**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add the following configuration items to the **StorageClass.yaml** file.

- Example 1: Configure zone and region information in the StorageClass.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: example-storageclass
provisioner: csi.huawei.com
parameters:
  volumeType: lun
  allocType: thin
volumeBindingMode: WaitForFirstConsumer
```



```
allowedTopologies:
- matchLabelExpressions:
  - key: topology.kubernetes.io/zone
    values:
    - ChengDu
  - key: topology.kubernetes.io/region
    values:
    - China-west
```

- Example 2: Configure protocol information in the StorageClass.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: protocol-example-storageclass
provisioner: csi.huawei.com
parameters:
  volumeType: lun
  allocType: thin
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
- matchLabelExpressions:
  - key: topology.kubernetes.io/protocol.iscsi
    values:
    - csi.huawei.com
```

Table 7-8 Parameter description

Parameter	Description	Remarks
volumeBindingMode	PersistentVolume binding mode, used to control the time when PersistentVolume resources are dynamically allocated and bound.	You can set this parameter to **WaitForFirstConsumer** or **Immediate**. **WaitForFirstConsumer**: indicates that the binding and allocation of the PersistentVolume are delayed until a Pod that uses the PVC is created. **Immediate**: The PersistentVolume is bound and allocated immediately after a PVC is created.
allowedTopologies/matchLabelExpressions	Topology information label, which is used to filter CSI backends and Kubernetes nodes. If the matching fails, PVCs or Pods cannot be created. Both **key** and **value** must be configured in a fixed format.	**key**: This parameter can be set to **topology.kubernetes.io/zone** or **topology.kubernetes.io/region**. **topology.kubernetes.io/protocol.**<*protocol*>: <*protocol*> indicates the protocol type and can be **iscsi**, **fc**, or **nfs**.
		value: If **key** is **topology.kubernetes.io/zone** or **topology.kubernetes.io/region**, **value** must be the same as the **topology label set in the prerequisites**. If **key** is **topology.kubernetes.io/protocol.**<*protocol*>, **value** is fixed to **csi.huawei.com**.

Step 7 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```

Step 8 Use the StorageClass to create a PVC with the topology capability. For details, see **6.2.1 Creating a PVC**.

Step 9 Use the PVC to create a Pod. For details, see **6.3.1 Creating a Pod**.

----End

7.6 Advanced Features of Enterprise Storage

7.6.1 Configuring QoS

This section describes how to create a LUN/file system volume that supports QoS.

Precautions

- The QoS feature is not a standard feature of Kubernetes and is customized by storage vendors.
- A QoS policy can be specified only when a StorageClass is created. Once the QoS policy is created, it cannot be modified because the StorageClass cannot be modified on Kubernetes.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

- Add the **qos** configuration item under **parameters**.
- The value of the **qos** section is JSON character strings in dictionary format. A character string is enclosed by single quotation marks and the dictionary key by double quotation marks. For details about the parameters, see **Table 7-9**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  qos: '{"IOTYPE": 2, "MINIOPS": 1000}'
```


Table 7-9 Parameters in qos

Parameter	Description	Remarks
IOTYPE	Read/write type.	This parameter is optional. If it is not specified, the default value of the storage backend is used. For details, see related storage documents. The value can be: • **0**: read I/O • **1**: write I/O • **2**: read and write I/Os
MAXBANDWIDTH	Maximum bandwidth.	The value is an integer greater than 0, expressed in MB/s.
MINBANDWIDTH	Minimum bandwidth.	The value is an integer greater than 0, expressed in MB/s.
MAXIOPS	Maximum IOPS.	The value is an integer greater than 0
MINIOPS	Minimum IOPS.	The value is an integer greater than 0
LATENCY	Maximum latency.	The value is an integer greater than 0, expressed in ms.

NOTE

- **MAXBANDWIDTH** or **MAXIOPS** cannot coexist with **MINBANDWIDTH**, **MINIOPS**, or **LATENCY**.
- For OceanStor Dorado, **IOTYPE** must be set to **2** (read and write I/Os), and **MINBANDWIDTH**, **MINIOPS**, and **LATENCY** are unavailable.
- vStore users do not support QoS policies.
- The QoS configuration takes effect only on the newly created PVC. QoS cannot be added automatically for PVCs with the same StorageClass name that have been provisioned.

Step 3 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```

Step 4 Use the StorageClass to create a PVC with the QoS capability. For details, see **6.2.1 Creating a PVC**.

----End

7.6.2 Configuring a vStore

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add the **vstoreName** parameter to the configuration of a backend. **vstoreName** indicates the vStore name on the storage device.

```
{
  "backends": [
  {
    ...
    "vstoreName": "***"
  }
  ]
}
```

NOTE

After configuring **huawei-csi-configmap.yaml**, restart huawei-csi-controller and huawei-csi-node. Otherwise, the configuration does not take effect.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

NOTE

When starting the huawei-csi services, enter the user name and password of the storage device vStore entered in **Step 3** in **4.7 Starting huawei-csi Services**.

----End

7.6.3 Configuring SAN Remote Replication

Perform this operation when you want to configure SAN remote replication.

Precautions

To use SAN remote replication, you need to configure a remote replication relationship between two storage devices in advance. For details, see the configuration guide of Huawei storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

In the **backends** section of the **huawei-csi-configmap.yaml** file, add two backends with a remote replication relationship and add the **replicaBackend** configuration item for each backend.

```
{
  "backends":[
  {
    ...
    "name": "replica1",
    "replicaBackend": "replica2"
  },
  {
    ...
    "name": "replica2",
    "replicaBackend": "replica1"
  }
  ]
}
```

NOTE

replicaBackend is the name of a peer end in remote replication. The two backends form a remote replication relationship. As shown in the preceding information, the peer end of **replica1** is **replica2**, and the peer end of **replica2** is **replica1**.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

Step 6 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and add related parameters in the **parameters** section of the .yaml file. For details about the parameters, see **Table 7-10**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  volumeType: lun
  replication: "true"
  replicationSyncPeriod: "3600"
  backend: "***"
```


Table 7-10 Parameter description

Parameter	Description	Remarks
replication	Whether a remote replication volume is to be created.	If this parameter is set to **true**, a remote replication volume is to be created. If this parameter is not set or set to **false**, no remote replication volume is to be created.
replicationSync-Period	Synchronization interval for remote replication.	This parameter is optional. The unit is second. The default value is 3600s.
backend	Name of the primary backend where the remote replication volume is to be created.	The value must be the same as the name of a remote replication backend configured in **huawei-csi-configmap.yaml**.

Step 7 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```

Step 8 Use the StorageClass to create a PVC with the SAN remote replication capability. For details, see **6.2.1 Creating a PVC**.

----End

7.6.4 Configuring SAN HyperMetro

Perform this operation when you want to configure SAN HyperMetro.

Precautions

To use SAN HyperMetro, you need to configure a HyperMetro relationship between two storage devices in advance. For details, see the configuration guide of Huawei storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

In the **backends** section of the **huawei-csi-configmap.yaml** file, add two backends with a HyperMetro relationship and add the **hyperMetroDomain** configuration item for each backend.

```
{
  "backends":[
  {
    ...
    "name": "hyperMetro1",
```



```
  "hyperMetroDomain": "***",
  "metroBackend": "hyperMetro2"
 },
 {
  ...
  "name": "hyperMetro2",
  "hyperMetroDomain": "***",
  "metroBackend": "hyperMetro1"
 }
 ]
}
```

NOTE

- **hyperMetroDomain** is the HyperMetro domain name configured between Huawei storage devices.
- **metroBackend** is the name of a peer end in HyperMetro. The two backends form a HyperMetro relationship. As shown in the preceding information, the peer end of **hyperMetro1** is **hyperMetro2**, and the peer end of **hyperMetro2** is **hyperMetro1**.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

Step 6 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and add related parameters in the **parameters** section of the .yaml file. For details about the parameters, see **Table 7-11**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add the following configuration items under **parameters** in the **StorageClass.yaml** file.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  volumeType: lun
  hyperMetro: "true"
```

Table 7-11 Parameter description

Parameter	Description	Remarks
hyperMetro	Whether a HyperMetro volume is to be created.	If this parameter is set to **true**, a HyperMetro volume is to be created. If this parameter is not set or set to **false**, no HyperMetro volume is to be created.

Step 7 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```


Step 8 Use the StorageClass to create a PVC with the SAN HyperMetro capability. For details, see **6.2.1 Creating a PVC**.

----End

7.6.5 Configuring NAS Remote Replication

Perform this operation when you want to configure NAS remote replication.

Precautions

To use NAS remote replication, you need to configure a remote replication relationship between two storage devices in advance. For details, see the configuration guide of Huawei storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

In the **backends** section of the **huawei-csi-configmap.yaml** file, add two backends with a remote replication relationship and add the **replicaBackend** configuration item for each backend.

```
{
  "backends":[
  {
    ...
    "name": "replica1",
    "replicaBackend": "replica2"
  },
  {
    ...
    "name": "replica2",
    "replicaBackend": "replica1"
  }
  ]
}
```

NOTE

replicaBackend is the name of a peer end in remote replication. The two backends form a remote replication relationship. As shown in the preceding information, the peer end of **replica1** is **replica2**, and the peer end of **replica2** is **replica1**.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

Step 6 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and add related parameters in the **parameters** section of the .yaml file. For details about the parameters, see **Table 7-12**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  volumeType: fs
  replication: "true"
  replicationSyncPeriod: "3600"
  backend: "***"
```

Table 7-12 Parameter description

Parameter	Description	Remarks
replication	Whether a remote replication volume is to be created.	If this parameter is set to **true**, a remote replication volume is to be created. If this parameter is not set or set to **false**, no remote replication volume is to be created.
replicationSync-Period	Synchronization interval for remote replication.	This parameter is optional. The unit is second. The default value is 3600s.
backend	Name of the primary backend where the remote replication volume is to be created.	This parameter is conditionally mandatory. In scenarios where vStores are used, if a vStore belongs to a remote replication vStore pair, the remote replication volume can be created only on the primary end of the vStore pair. Therefore, you need to use the **backend** parameter to specify the name of the primary end on which the remote replication volume is to be created, which must be the same as the name of a remote replication backend configured in **huawei-csi-configmap.yaml**.

Step 7 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```

Step 8 Use the StorageClass to create a PVC with the NAS remote replication capability. For details, see **6.2.1 Creating a PVC**.

----End

7.6.6 Configuring NAS HyperMetro

Perform this operation when you want to configure NAS HyperMetro.

Precautions

To use NAS HyperMetro, you need to configure a HyperMetro relationship between two storage devices and create a HyperMetro vStore pair in advance. For details, see the configuration guide of Huawei storage.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi huawei-csi-configmap.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

In the **backends** section of the **huawei-csi-configmap.yaml** file, add two backends with a HyperMetro relationship and add the **metrovStorePairID** configuration item for each backend.

```
{
  "backends":[
  {
    ...
    "name": "hyperMetro1",
    "vstoreName": "***",
    "metrovStorePairID": "***",
    "metroBackend": "hyperMetro2"
  },
  {
    ...
    "name": "hyperMetro2",
    "vstoreName": "***",
    "metrovStorePairID": "***",
    "metroBackend": "hyperMetro1"
  }
  ]
}
```

NOTE

- NAS HyperMetro is supported only when a vStore is configured. For details about how to configure a vStore, see **7.6.2 Configuring a vStore**.
- **metrovStorePairID** is the ID of the HyperMetro vStore pair to which a vStore belongs.
- **metroBackend** is the name of a peer end in HyperMetro. The two backends form a HyperMetro relationship. As shown in the preceding information, the peer end of **hyperMetro1** is **hyperMetro2**, and the peer end of **hyperMetro2** is **hyperMetro1**.

Step 3 Run the **kubectl create -f huawei-csi-configmap.yaml** command to create **huawei-csi-configmap**.

```
# kubectl create -f huawei-csi-configmap.yaml
```

Step 4 After the creation is complete, run the **kubectl get configmap -n kube-system | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

```
# kubectl get configmap -n kube-system | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.7 Starting huawei-csi Services**.

Step 6 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and add the **hyperMetro** parameter under **parameters**

in the .yaml file. For details about the parameters, see **Table 7-13**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  volumeType: fs
  hyperMetro: "true"
```

Table 7-13 Parameter description

Parameter	Description	Remarks
hyperMetro	Whether a HyperMetro volume is to be created.	If this parameter is set to **true**, a HyperMetro volume is to be created. If this parameter is not set or set to **false**, no HyperMetro volume is to be created.

Step 7 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```

Step 8 Use the StorageClass to create a PVC with the NAS HyperMetro capability. For details, see **6.2.1 Creating a PVC**.

----End

7.6.7 Configuring an Application Type

This section describes how to create a LUN/file system volume that supports different application types.

Precautions

- The application type feature is not a standard feature of Kubernetes and is customized by storage vendors.
- An application type can be specified only when a PVC is created.
- A created PVC cannot be modified on Kubernetes.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

Add the **applicationType** configuration item under **parameters**. The value of **applicationType** is a character string. For details, see **Table 7-14**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
```



```
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  volumeType: "***"
  applicationType: "***"
```

Table 7-14 Parameter description of applicationType

Parameter	Description	Remarks
applicationType	Application type name on the storage device. The value is a character string.	If the value of **volumeType** is **lun**, log in to DeviceManager and choose **Services** > **Block Service** > **LUN Groups** > **LUNs** > **Create** to obtain the application type name. If the value of **volumeType** is **fs**, log in to DeviceManager and choose **Services** > **File Service** > **File Systems** > **Create** to obtain the application type name.

NOTE

This feature applies only to OceanStor Dorado V6 series storage systems.

Step 3 Run the following command to create a StorageClass based on the .yaml file.

```
# kubectl create -f StorgeClass.yaml
```

Step 4 Use the StorageClass to create a PVC with the application type capability. For details, see **6.2.1 Creating a PVC**.

----End

7.7 Advanced Features of Distributed Storage

7.7.1 Configuring QoS

This section describes how to create a LUN volume that supports QoS.

Precautions

- The QoS feature is not a standard feature of Kubernetes and is customized by storage vendors.
- A QoS policy can be specified only when a StorageClass is created. Once the QoS policy is created, it cannot be modified because the StorageClass cannot be modified on Kubernetes.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **vi StorageClass.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

- Add the **qos** configuration item under **parameters**.
- The value of the **qos** section is JSON character strings in dictionary format. A character string is enclosed by single quotation marks and the dictionary key by double quotation marks. For details about the parameters, see **Table 7-15**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "***"
provisioner: "csi.huawei.com"
parameters:
  ...
  qos: '{"maxMBPS": 999, "maxIOPS": 999}'
```

Table 7-15 Parameters in qos

Parameter	Description	Remarks
maxMBPS	Maximum bandwidth.	This parameter is mandatory. The value is an integer greater than 0, expressed in MB/s.
maxIOPS	Maximum IOPS.	This parameter is mandatory. The value is an integer greater than 0

Step 3 Run the following command to create a PVC based on the .yaml file.

```
# kubectl create -f StorageClass.yaml
```

----End

8 Common Operations

8.1 Uninstalling CSI

Perform this operation when you want to uninstall CSI.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-node service (**huawei-csi-node.yaml** is the configuration information in **Step 8** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-node.yaml
```

Step 3 Run the following command to view the huawei-csi-node service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-node
```

Step 4 Run the following command to delete the huawei-csi-controller service (**huawei-csi-controller.yaml** is the configuration information in **Step 6** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-controller.yaml
```

Step 5 Run the following command to view the huawei-csi-controller service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-controller
```


Step 6 Run the following command to delete the RBAC permission (**huawei-csi-rbac.yaml** is the configuration information in **Step 4** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-rbac.yaml
```

Step 7 Run the following command to delete the huawei-csi-configmap object. By default, the **<namespace>** parameter of the huawei-csi-configmap object is **kube-system**. Replace it with the actual name.

```
# kubectl delete configmap huawei-csi-configmap -n <namespace>
```

Step 8 Run the following command to view the **configmap** information of huawei-csi. If no command output is displayed, the deletion is complete.

```
# kubectl get configmap -A | grep huawei-csi-configmap
```

Step 9 Run the following command to delete the huawei-csi-secret object. By default, the **<namespace>** parameter of the huawei-csi-secret object is **kube-system**. Replace it with the actual name.

```
# kubectl delete secret huawei-csi-secret -n <namespace>
```

Step 10 Run the following command to view the **secret** information of huawei-csi. If no command output is displayed, the deletion is complete.

```
# kubectl get secret -A | grep huawei-csi-secret
```

----End

8.2 Updating the User Name or Password of a Storage Device Configured on CSI

When the user name or password of a storage device changes, you need to update the configuration information on CSI. Otherwise, huawei-csi services cannot work properly.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **chmod +x secretUpdate** command to grant the execute permission on the secretUpdate tool.

Step 3 Run the **./secretUpdate** command to start the secretUpdate tool and enter related information as prompted.

```
# ./secretUpdate
**************************All Secret Info*****************************
<current secret info>
**********************************************************************
**************************All Backend Info****************************
The 1 backend name is: <backend-1-name>   backend url is: [<backend-1-url>]
Do you want to update it? Y/N               # Specify whether to update information.
Enter backend <backend-1-name>'s user:      # Enter the user name of storage device 1.
Enter backend <backend-1-name>'s password:  # Enter the password of storage device 1.
Please enter the password again:            # Enter the password of storage device 1 again.

The 2 backend name is: <backend-2-name>   backend url is: [<backend-2-url>]
Do you want to update it? Y/N               # Specify whether to update information.
Enter backend <backend-2-name>'s user:      # Enter the user name of storage device 2.
Enter backend <backend-2-name>'s password:  # Enter the password of storage device 2.
```



```
Please enter the password again:            # Enter the password of storage device 2 again.

**********Update CSI Secret Successful**********
```

Step 4 Run the following command to restart the huawei-csi-controller service.

```
# kubectl get deployment huawei-csi-controller -o yaml --namespace=kube-system | kubectl replace --force
-f -
```

Step 5 Run the following command to restart the huawei-csi-node service.

```
# kubectl get daemonset huawei-csi-node -o yaml --namespace=kube-system | kubectl replace --force -f -
```

Step 6 Run the **kubectl get pod -A | grep huawei** command to check whether the services are restarted successfully.

```
# kubectl get pod -A | grep huawei
kube-system   huawei-csi-controller-695b84b4d8-tg64l   3/3     Running     0          14s
kube-system   huawei-csi-node-g6f7z                    2/2     Running     0          14s
```

----End

8.3 Updating the configmap Object of huawei-csi

Perform this operation when you want to add a storage pool to an existing backend or change an existing service IP address.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **kubectl edit configmap huawei-csi-configmap -n kube-system** command, press **I** or **Insert** to enter the editing mode, and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-san",
                "name": "storage",
                "urls": ["https://192.168.125.20:8088", "https://192.168.125.21:8088"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "iscsi", "portals": ["192.168.125.22", "192.168.125.23"]}
            }
        ]
    }
```


Table 8-1 Description of configuration items

Configuration Item	Format	Description	Remarks
name	String	User-defined name of a storage device.	• User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). • This parameter cannot be modified.
storage	String	Type of the storage device to be connected.	• In the scenario where the enterprise storage SAN is connected, the value is fixed to **oceanstor-san**. • This parameter cannot be modified.
pools	List	Name of a storage pool used on the storage device to be connected.	• One or more storage pools on the same storage device are supported. Use commas (,) to separate multiple storage pools. • Currently, only storage pools can be added. • You can log in to DeviceManager to obtain the storage pools that support the block storage service.
urls	List	Management URL of the storage device to be connected.	One or more management URLs of the same storage device are supported. Use commas (,) to separate multiple management URLs. Currently, only IPv4 addresses are supported. Example: https://192.168.125.20:8088 **NOTE** A storage device has multiple controllers, and each controller has a management URL. Therefore, a storage device has multiple management URLs.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in scenarios where the iSCSI protocol is used.	In scenarios where the iSCSI protocol is used, set the **protocol** parameter to a fixed value: **iscsi**. Set the **portals** parameter to the iSCSI service IP addresses of the storage device. Use commas (,) to separate multiple iSCSI service IP addresses. You can log in to DeviceManager to obtain the iSCSI service IP addresses. ● OceanStor V3/V5 series: To obtain the IP address of an Ethernet port on a specified storage device, choose **Provisioning** > **Port** > **Ethernet Ports** on DeviceManager. ● OceanStor Dorado V3 series: To obtain the IP address of an Ethernet port on a specified storage device, choose **Provisioning** > **Port** > **Ethernet Ports** on DeviceManager. ● OceanStor Dorado V6 series: To obtain the IP address of a logical port on a specified storage device, choose **Services** > **Network** > **Logical Ports** on DeviceManager and obtain the IP address whose data protocol is iSCSI.

Step 3 Run the following command to restart the huawei-csi-controller service.

```
# kubectl get deployment huawei-csi-controller -o yaml --namespace=kube-system | kubectl replace --force
-f -
```

Step 4 Run the following command to restart the huawei-csi-node service.

```
# kubectl get daemonset huawei-csi-node -o yaml --namespace=kube-system | kubectl replace --force -f -
```

Step 5 Run the **kubectl get pod -A | grep huawei** command to check whether the services are restarted successfully.

```
# kubectl get pod -A | grep huawei
kube-system   huawei-csi-controller-695b84b4d8-tg64l   3/3   Running   0   14s
kube-system   huawei-csi-node-g6f7z                    2/2   Running   0   14s
```

----End

8.4 Adding a Backend for huawei-csi

Perform this operation when you want to add a storage device or a storage pool as an independent backend.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Configure multiple backends. For details, see **7.1 Configuring Multiple Backends**.

Step 3 Run the **chmod +x secretUpdate** command to grant the execute permission on the secretUpdate tool.

Step 4 Run the **./secretUpdate** command to start the secretUpdate tool and enter related information as prompted.

```
# ./secretUpdate
*************************All Secret Info*****************************
<current secret info>
*********************************************************************
*************************All Backend Info*****************************
The 1 backend name is: <backend-1-name>   backend url is: [<backend-1-url>]
Do you want to update it? Y/N               # Specify whether to update information.
Enter backend <backend-1-name>'s user:      # Enter the user name of storage device 1.
Enter backend <backend-1-name>'s password:  # Enter the password of storage device 1.
Please enter the password again:            # Enter the password of storage device 1 again.

The 2 backend name is: <backend-2-name>   backend url is: [<backend-2-url>]
Do you want to update it? Y/N               # Specify whether to update information.
Enter backend <backend-2-name>'s user:      # Enter the user name of storage device 2.
Enter backend <backend-2-name>'s password:  # Enter the password of storage device 2.
Please enter the password again:            # Enter the password of storage device 2 again.

********************Update CSI Secret Successful*********************
```

Step 5 Run the following command to restart the huawei-csi-controller service.

```
# kubectl get deployment huawei-csi-controller -o yaml --namespace=kube-system | kubectl replace --force
-f -
```

Step 6 Run the following command to restart the huawei-csi-node service.

```
# kubectl get daemonset huawei-csi-node -o yaml --namespace=kube-system | kubectl replace --force -f -
```

Step 7 Run the **kubectl get pod -A | grep huawei** command to check whether the services are restarted successfully.

```
# kubectl get pod -A | grep huawei
kube-system   huawei-csi-controller-695b84b4d8-tg64l   3/3   Running   0   14s
kube-system   huawei-csi-node-g6f7z                    2/2   Running   0   14s
```

----End

8.5 Updating the huawei-csi-controller Service

Perform this operation when you need to update the huawei-csi-controller service, for example, adding the snapshot or the capacity expansion function.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-controller service (**huawei-csi-controller.yaml** is the configuration information in **Step 6** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-controller.yaml
```

Step 3 Run the following command to view the huawei-csi-controller service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-controller
```

Step 4 Run the following command to delete the RBAC permission (**huawei-csi-rbac.yaml** is the configuration information in **Step 4** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-rbac.yaml
```

Step 5 Run the **vi huawei-csi-rbac.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. (For details, see sample file **yamls/huawei-csi-rbac.yaml** in the software package.)

NOTE

- The csi-resizer service is supported since Kubernetes v1.16.

  For details about the **huawei-csi-rbac.yaml** file, see sample file **yamls/huawei-csi-resize-rbac.yaml** in the software package.
- The csi-snapshotter service is supported since Kubernetes v1.17.

  For details about the **huawei-csi-rbac.yaml** file, see sample file **yamls/huawei-csi-resize-snapshot-rbac.yaml** in the software package.

Step 6 Run the following command to create the RBAC permission.

```
# kubectl create -f huawei-csi-rbac.yaml
```

Step 7 Run the **vi huawei-csi-controller.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. (For details, see sample file **yamls/huawei-csi-controller.yaml** in the software package.)

NOTE

- In the **huawei-csi:****.*.** configuration item in the sample .yaml file, **.*.** must be replaced with the version number of the created Huawei CSI image.
- The csi-resizer service is supported since Kubernetes v1.16.

  For details about the **huawei-csi-controller.yaml** file, see sample file **yamls/huawei-csi-resize-controller.yaml** in the software package.
- The csi-snapshotter service is supported since Kubernetes v1.17.

  For details about the **huawei-csi-controller.yaml** file, see sample file **yamls/huawei-csi-resize-snapshot-controller.yaml** in the software package.

Step 8 Run the following command to start the controller service.

```
# kubectl create -f huawei-csi-controller.yaml
```

Step 9 After the huawei-csi service is deployed, run the **kubectl get pod -A | grep huawei-csi-controller** command to check whether the service is started.


```
# kubectl get pod -A | grep huawei-csi-controller
kube-system   huawei-csi-controller-695b84b4d8-tg64l   3/3     Running     0          14s
```

----End

8.6 Updating the huawei-csi-node Service

Perform this operation when you need to update the huawei-csi-node service.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-node service (**huawei-csi-node.yaml** is the configuration information in **Step 8** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-node.yaml
```

Step 3 Run the following command to view the huawei-csi-node service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-node
```

Step 4 Run the **vi huawei-csi-node.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. Compile the **huawei-csi-node.yaml** file. For details, see sample file **yamls/huawei-csi-node.yaml** in the software package.

- In the **huawei-csi:****.*.** configuration item in the sample .yaml file, **.*.** must be replaced with the version number of the created Huawei CSI image.
- In the **args** parameter in the **huawei-csi-driver** section in the .yaml file, **--volume-use-multipath** indicates that multipathing is enabled by default. The following shows how to change the value.

```
args:
  - "--endpoint=/csi/csi.sock"
  - "--containerized"
  - "--driver-name=csi.huawei.com"
  - "--volume-use-multipath=false"
```

Step 5 Run the following command to start the node service.

```
# kubectl create -f huawei-csi-node.yaml
```

Step 6 After the huawei-csi service is deployed, run the **kubectl get pod -A | grep huawei-csi-node** command to check whether the service is started.

```
# kubectl get pod -A | grep huawei-csi-node
kube-system huawei-csi-node-g6f7z 2/2 Running 0 14s
```

----End

8.7 Modifying the Log Output Mode

huawei-csi supports two log output modes: **file** and **console**. **file** indicates that logs are output to the fixed directory (**/var/log/huawei**), and **console** indicates that logs are output to the standard directory of the container. You can set the log output mode as required. The default mode is **file**.

8.7.1 Modifying the Log Output Mode of the huawei-csi-controller Service

Perform this operation when you want to set the log output mode of the huawei-csi-controller service.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-controller service (**huawei-csi-controller.yaml** is the configuration information in **Step 6** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-controller.yaml
```

Step 3 Run the following command to view the huawei-csi-controller service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-controller
```

Step 4 Run the **vi huawei-csi-controller.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. (For details, see sample file **yamls/huawei-csi-controller.yaml** in the software package.) For details about the parameters, see **Table 8-2**.

```
args:
  - "--endpoint=$(CSI_ENDPOINT)"
  - "--controller"
  - "--containerized"
  - "--driver-name=csi.huawei.com"
  - "--loggingModule=file"
  - "--logLevel=info"
  - "--logFileDir=/var/log/huawei"
  - "--logFileSize=20M"
  - "--maxBackups=9"
```

Table 8-2 Description of log output parameters

Configuration Item	Description	Remarks
loggingModule	huawei-csi log output mode.	The value can be **file** or **console**. The default value is **file**.
logLevel	huawei-csi log output level.	Supported levels are **debug**, **info**, **warning**, **error**, and **fatal**. The default level is **info**.
logFileDir	huawei-csi log directory in **file** output mode.	This parameter is available only when **loggingModule** is set to **file**. The default log directory is **/var/log/huawei**.
logFileSize	Size of a single huawei-csi log file in **file** output mode.	This parameter is available only when **loggingModule** is set to **file**. The default log file size is 20 MiB.

Configuration Item	Description	Remarks
maxBackups	Maximum number of huawei-csi log file backups in **file** output mode.	This parameter is available only when **loggingModule** is set to **file**. The default number of log file backups is 9.

Step 5 Run the following command to start the controller service.

```
# kubectl create -f huawei-csi-controller.yaml
```

Step 6 After the huawei-csi service is deployed, run the **kubectl get pod -A | grep huawei-csi-controller** command to check whether the service is started.

```
# kubectl get pod -A -o wide | grep huawei
kube-system   huawei-csi-controller-b59577886-qqzm8   3/3     Running     0          18h   10.244.1.67
node     <none>           <none>
```

Step 7 View the logs of the huawei-csi-controller service.

- If **loggingModule** is set to **file**, log in to the node, go to the log directory specified by **logFileDir**, and run the following command to view the log of huawei-csi-controller.
  ```
  # tail -f huawei-csi-controller
  ```
- If **loggingModule** is set to **console**, run the following command to view the log of huawei-csi-controller.
  ```
  # kubectl logs huawei-csi-controller -c huawei-csi-driver -n kube-system
  ```

----End

8.7.2 Modifying the Log Output Mode of the huawei-csi-node Service

Perform this operation when you want to set the log output mode of the huawei-csi-node service.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-node service (**huawei-csi-node.yaml** is the configuration information in **Step 8** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-node.yaml
```

Step 3 Run the following command to view the huawei-csi-node service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-node
```

Step 4 Run the **vi huawei-csi-node.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. Compile the **huawei-csi-node.yaml** file. For details, see sample file **yamls/**

huawei-csi-node.yaml in the software package. For details about the parameters, see **Table 8-3**.

```
args:
  - "--endpoint=/csi/csi.sock"
  - "--containerized"
  - "--driver-name=csi.huawei.com"
  - "--volume-use-multipath=false"
  - "--loggingModule=file"
  - "--logLevel=info"
  - "--logFileDir=/var/log/huawei"
  - "--logFileSize=20M"
  - "--maxBackups=9"
```

Table 8-3 Description of log output parameters

Configuration Item	Description	Remarks
loggingModule	huawei-csi log output mode.	The value can be **file** or **console**. The default value is **file**.
logLevel	huawei-csi log output level.	Supported levels are **debug**, **info**, **warning**, **error**, and **fatal**. The default level is **info**.
logFileDir	huawei-csi log directory in **file** output mode.	This parameter is available only when **loggingModule** is set to **file**. The default log directory is **/var/log/huawei**.
logFileSize	Size of a single huawei-csi log file in **file** output mode.	This parameter is available only when **loggingModule** is set to **file**. The default log file size is 20 MiB.
maxBackups	Maximum number of huawei-csi log file backups in **file** output mode.	This parameter is available only when **loggingModule** is set to **file**. The default number of log file backups is 9.

Step 5 Run the following command to start the node service.

```
# kubectl create -f huawei-csi-node.yaml
```

Step 6 After the huawei-csi service is deployed, run the **kubectl get pod -A | grep huawei-csi-node** command to check whether the service is started.

```
# kubectl get pod -A | grep huawei-csi-node
kube-system   huawei-csi-node-4sfwr                2/2     Running            0          18h  10.244.1.68
node     <none>           <none>
```

Step 7 View the logs of the huawei-csi-node service.

- If **loggingModule** is set to **file**, log in to the node, go to the log directory specified by **logFileDir**, and run the following command to view the log of huawei-csi-node.

  ```
  # tail -f huawei-csi-node
  ```

- If **loggingModule** is set to **console**, run the following command to view the log of huawei-csi-node.

  ```
  # kubectl logs huawei-csi-node -c huawei-csi-driver -n kube-system
  ```

----End

9 FAQ

9.1 Viewing Log Information

Viewing Logs Generated When the secret Object Is Configured

Step 1 Run the **cd /var/log/huawei** command to go to the log directory.

```
# cd /var/log/huawei
```

Step 2 Run the following command to view the logs of huawei-csi-install.

```
# vi huawei-csi-install
```

----End

Viewing Logs of the huawei-csi-controller Service

Step 1 Run the following command to obtain the node where huawei-csi-controller is located.

```
# kubectl get pod -A -o wide | grep huawei
kube-system   huawei-csi-controller-695b84b4d8-tg64l   3/3   Running   0   14s   <host1-ip>   <host1-name>   <none>   <none>
```


Step 2 Use a remote access tool, such as PuTTY, to log in to the huawei-csi-controller node in the Kubernetes cluster through the management IP address.

Step 3 Run the **cd /var/log/huawei** command to go to the log directory.

```
# cd /var/log/huawei
```

Step 4 Run the following command to view the customized output logs of the container.

```
# vi huawei-csi-controller
```

Step 5 Run the **cd /var/log/containers** command to go to the container directory.

```
# cd /var/log/containers
```

Step 6 Run the following command to view the standard output logs of the container.

```
# vi huawei-csi-controller-<name>_kube-system_huawei-csi-driver-<contrainer-id>.log
```

----End

Viewing Logs of the huawei-csi-node Service

Step 1 Run the following command to obtain the node where huawei-csi-node is located.

```
# kubectl get pod -A -o wide | grep huawei
kube-system   huawei-csi-node-g6f7z                  2/2     Running     0          14s   <host2-ip>    <host2-name>   <none>           <none>
```

Step 2 Use a remote access tool, such as PuTTY, to log in to the huawei-csi-node node in the Kubernetes cluster through the management IP address.

Step 3 Run the **cd /var/log/huawei** command to go to the log directory.

```
# cd /var/log/huawei
```

Step 4 Run the following command to view the customized output logs of the container.

```
# vi huawei-csi-node
```

Step 5 Run the **cd /var/log/containers** command to go to the container directory.

```
# cd /var/log/containers
```

Step 6 Run the following command to view the standard output logs of the container.

```
# vi huawei-csi-node-<name>_kube-system_huawei-csi-driver-<contrainer-id>.log
```

----End

9.2 Failed to Create a Pod Because the iscsi_tcp Service Is Not Started Properly When the Kubernetes Platform Is Set Up for the First Time

Symptom

When you create a Pod, error **Cannot connect ISCSI portal *.*.*.*: libkmod: kmod_module_insert_module: could not find module by name='iscsi_tcp'** is reported in the **/var/log/huawei-csi-node** log.

Environment Configuration

Kubernetes version: 1.13 or later

Root Cause Analysis

The iscsi_tcp service may be stopped after the Kubernetes platform is set up and the iscsi service is installed. You can run the **lsmod | grep iscsi | grep iscsi_tcp** command to check whether the service is stopped.

```
# lsmod | grep iscsi | grep iscsi_tcp
iscsi_tcp              18333  6
libiscsi_tcp           25146  1 iscsi_tcp
libiscsi               57233  2 libiscsi_tcp,iscsi_tcp
scsi_transport_iscsi   99909  3 iscsi_tcp,libiscsi
```

Solution or Workaround

Run the **modprobe iscsi_tcp** command to manually load the iscsi_tcp service.

```
# modprobe iscsi_tcp
# lsmod | grep iscsi | grep iscsi_tcp
iscsi_tcp              18333  6
libiscsi_tcp           25146  1 iscsi_tcp
```

9.3 Failed to Start the huawei-csi-node Service with Error Message "/var/lib/iscsi is not a directory" Reported

Symptom

The huawei-csi-node service cannot be started. When you run the **kubectl describe daemonset huawei-csi-node -n kube-system** command, error message "/var/lib/iscsi is not a directory" is reported.

Environment Configuration

Kubernetes version: 1.13 or later

Root Cause Analysis

The **/var/lib/iscsi** directory does not exist in the huawei-csi-node container.

Solution or Workaround

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to delete the huawei-csi-node service (**huawei-csi-node.yaml** is the configuration information in **Step 8** in **4.7 Starting huawei-csi Services**).

```
# kubectl delete -f huawei-csi-node.yaml
```

Step 3 Run the following command to view the huawei-csi-node service. If no command output is displayed, the deletion is complete.

```
# kubectl get pod -A | grep huawei-csi-node
```

Step 4 Run the **vi huawei-csi-node.yaml** command to modify the .yaml file. Press **I** or **Insert** to enter the editing mode, set **path** in **huawei-csi-node.yaml** > **volumes** >

iscsi-dir > **hostPath** to **/var/lib/iscsi** and delete the **type** line. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. Compile the **huawei-csi-node.yaml** file. For details, see sample file **yamls/huawei-csi-node.yaml** in the software package.

Step 5 Run the following command to start the node service.

```
# kubectl create -f huawei-csi-node.yaml
```

Step 6 After the huawei-csi service is deployed, run the **kubectl get pod -A | grep huawei-csi-node** command to check whether the service is started.

```
# kubectl get pod -A | grep huawei-csi-node
kube-system huawei-csi-node-g6f7z 2/2 Running 0 14s
```

----End

9.4 After a Worker Node in the Cluster Breaks Down and Recovers, Pod Failover Is Complete but the Source Host Where the Pod Resides Has Residual Drive Letters

Symptom

A Pod is running on worker node A, and an external block device is mounted to the Pod through CSI. After worker node A is powered off abnormally, the Kubernetes platform detects that the node is faulty and switches the Pod to worker node B. After worker node A recovers, the drive letters on worker node A change from normal to faulty.

Environment Configuration

Kubernetes version: 1.13 or later

Storage type: block storage

Root Cause Analysis

After worker node A recovers, Kubernetes initiates an unmapping operation on the storage, but does not initiate a drive letter removal operation on the host. After Kubernetes completes the unmapping, residual drive letters exist on worker node A.

Solution or Workaround

Currently, you can only manually clear the residual drive letters on the host. Alternatively, restart the host again and use the disk scanning mechanism during the host restart to clear the residual drive letters. The specific method is as follows:

Troubleshooting method:

Step 1 Run the **multipath -ll** command to check whether a DM multipathing device with abnormal multipathing status exists.

As shown in the following figure, the path status is **failed faulty running**, the corresponding DM multipathing device is **dm-12**, and the associated SCSI disks are

sdi and **sdj**. If multiple paths are configured, multiple SCSI disks exist. Record these SCSI disks.

```
# multipath -ll
mpathb (3618cf24100f8f457014a764c000001f6) dm-12 HUAWEI  ,XSG1
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=-1 status=active
  |- 39:0:0:1        sdi 8:48  failed faulty running
  `- 38:0:0:1        sdj 8:64  failed faulty running
```

- If yes, go to **Step 3**.
- If no, no further action is required.

Step 2 Check whether the residual DM multipathing device is readable.

Run the **dd if=/dev/*dm-xx* of=/dev/null count=1 bs=1M iflag=direct** command.

dm-xx indicates the device ID obtained in **Step 1**.

If the returned result is **Input/output error** and the read data is **0 bytes (0 B) copied**, the device is unreadable.

```
#dd if=/dev/dm-12 of=/dev/null count=1 bs=1M iflag=direct
dd: error reading '/dev/dm-12': Input/output error
0+0 records in
0+0 records out
0 bytes (0 B) copied, 0.0236862 s, 0.0 kB/s
```

- If yes, record the residual *dm-xx* device and associated disk IDs (for details, see **Step 1**) and perform the clearing operation.
- If the command execution is suspended, go to **3**.
- If other cases, contact technical support engineers.

Step 3 Log in to the node again in another window.

1. Run the following command to view the suspended process.
   ```
   # ps -ef | grep dm-12 | grep -w dd
   root     21725  9748  0 10:33 pts/10   00:00:00 dd if=/dev/dm-12 of=/dev/null count=1 bs=10M
   iflag=direct
   ```
2. Kill the pid.
   ```
   # kill -9 pid
   ```
3. Record the residual *dm-xx* device and associated disk IDs (for details, see **Step 1**) and perform the clearing operation.

----End

Cleanup procedure:

Step 1 Run the **multipath -f /dev/*dm-xx*** command to delete residual multipathing aggregation device information according to the DM multipathing device obtained in the troubleshooting method.

```
# multipath -f /dev/dm-12
```

If an error is reported, contact technical support engineers.

Step 2 Run the following command to clear the residual SCSI disks according to the drive letters of the residual disks obtained in the troubleshooting method.

echo 1 > /sys/block/*xxxx*/device/delete

When multiple paths are configured, clear the residual disks based on the drive letters. The residual paths are **sdi** and **sdj**.


```
# echo 1 > /sys/block/sdi/device/delete
# echo 1 > /sys/block/sdj/device/delete
```

If an error is reported, contact technical support engineers.

Step 3 Check whether the DM multipathing device and SCSI disk information has been cleared.

Run the **multipath -ll**, **ls -l /sys/block/**, and **ls -l /dev/disk/by-id/** commands in sequence to query the path and disk information. If the residual **dm-12** device and SCSI disks **sdi** and **sdj** are cleared, the clearing is complete.

```
# multipath -ll
mpathb (3618cf24100f8f457014a764c000001f6) dm-3 HUAWEI  ,XSG1
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=-1 status=active
  |- 39:0:0:1        sdd 8:48  active ready running
  `- 38:0:0:1        sde 8:64  active ready running
mpathn (3618cf24100f8f457315a764c000001f6) dm-5 HUAWEI  ,XSG1
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=-1 status=active
  |- 39:0:0:2        sdc 8:32  active ready running
  `- 38:0:0:2        sdb 8:16  active ready running
# ls -l /sys/block/
total 0
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-0 -> ../devices/virtual/block/dm-0
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-1 -> ../devices/virtual/block/dm-1
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-2 -> ../devices/virtual/block/dm-2
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-3 -> ../devices/virtual/block/dm-3
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdb -> ../devices/platform/host35/session2/target35:0:0/35:0:0:1/
block/sdb
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdc -> ../devices/platform/host34/
target34:65535:5692/34:65535:5692:0/block/sdc
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdd -> ../devices/platform/host39/session6/target39:0:0/39:0:0:1/
block/sdd
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sde -> ../devices/platform/host38/session5/target38:0:0/38:0:0:1/
block/sde
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdh -> ../devices/platform/host39/session6/target39:0:0/39:0:0:3/
block/sdh
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdi -> ../devices/platform/host38/session5/target38:0:0/38:0:0:3/
block/sdi
ls -l /dev/disk/by-id/
total 0
lrwxrwxrwx 1 root root 10 Aug 11 19:57 dm-name-mpathb -> ../../dm-3
lrwxrwxrwx 1 root root 10 Aug 11 19:58 dm-name-mpathn -> ../../dm-5
lrwxrwxrwx 1 root root 10 Aug 11 19:57 dm-uuid-mpath-3618cf24100f8f457014a764c000001f6 -> ../../dm-3
lrwxrwxrwx 1 root root 10 Aug 11 19:58 dm-uuid-mpath-3618cf24100f8f457315a764c000001f6 -> ../../dm-5
lrwxrwxrwx 1 root root  9 Aug 11 19:57 scsi-3618cf24100f8f457014a764c000001f6 -> ../../sdd
lrwxrwxrwx 1 root root  9 Aug 11 19:57 scsi-3618cf24100f8f45712345678000103e8 -> ../../sdi
lrwxrwxrwx 1 root root  9 Aug  3 15:17 scsi-3648435a100588052786543 21ffffffff -> ../../sdb
lrwxrwxrwx 1 root root  9 Aug  2 14:49 scsi-368886030000020aff44cc0d060c987f1 -> ../../sdc
lrwxrwxrwx 1 root root  9 Aug 11 19:57 wwn-0x618cf24100f8f457014a764c000001f6 -> ../../sdd
lrwxrwxrwx 1 root root  9 Aug 11 19:57 wwn-0x618cf24100f8f45712345678000103e8 -> ../../sdi
lrwxrwxrwx 1 root root  9 Aug  3 15:17 wwn-0x648435a100588052786543 21ffffffff -> ../../sdb
lrwxrwxrwx 1 root root  9 Aug  2 14:49 wwn-0x68886030000020aff44cc0d060c987f1 -> ../../sdc
```

----End

9.5 Failed to Start huawei-csi Services with the Status Displayed as InvalidImageName

Symptom

The huawei-csi services (huawei-csi-controller or huawei-csi-node) cannot be started. After the **kubectl get pod -A | grep huawei** command is executed, the command output shows that the service status is **InvalidImageName**.

```
# kubectl get pod -A | grep huawei
kube-system     huawei-csi-controller-fd5f97768-qlldc     5/6     InvalidImageName     0          16s
kube-system     huawei-csi-node-25txd                     1/2     InvalidImageName     0          15s
```

Environment Configuration

Kubernetes version: 1.13 or later

Root Cause Analysis

In **huawei-csi-controller.yaml** and **huawei-csi-node.yaml**, the Huawei CSI image version number is incorrect. For example:

```
...
- name: huawei-csi-driver
  image: huawei-csi:*.*.*
...
```

Solution or Workaround

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to modify the configuration file of the huawei-csi-node service. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
# kubectl edit daemonset huawei-csi-node -o yaml --namespace=kube-system
```

NOTE

In the **huawei-csi:**.*.**** configuration item in the sample .yaml file, **.*.** must be replaced with the version number of the created Huawei CSI image.

Step 3 Run the following command to modify the configuration file of the huawei-csi-controller service: Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
# kubectl edit deployment huawei-csi-controller -o yaml --namespace=kube-system
```

NOTE

In the **huawei-csi:**.*.**** configuration item in the sample .yaml file, **.*.** must be replaced with the version number of the created Huawei CSI image.

Step 4 Wait until the huawei-csi-node and huawei-csi-controller services are started.

Step 5 Run the following command to check whether the huawei-csi services are started.

```
# kubectl get pod -A  | grep huawei
kube-system   huawei-csi-controller-58799449cf-zvhmv   6/6     Running       0          2m29s
kube-system   huawei-csi-node-7fxh6                    2/2     Running       0          12m
```

----End

9.6 When a PVC Is Created, the PVC Is in the Pending State

Symptom

A PVC is created. After a period of time, the PVC is still in the **Pending** state.

Environment Configuration

Kubernetes version: 1.13 or later

Root Cause Analysis

Cause 1: A StorageClass with the specified name is not created in advance. As a result, Kubernetes cannot find the specified StorageClass name when a PVC is created.

Cause 2: The storage pool capability does not match the StorageClass capability. As a result, huawei-csi fails to select a storage pool.

Cause 3: An error code (for example, 50331651) is returned by a RESTful interface of the storage. As a result, huawei-csi fails to create a PVC.

Cause 4: The storage does not return a response within the timeout period set by huawei-csi. As a result, huawei-csi returns a timeout error to Kubernetes.

Cause 5: Other causes.

Solution or Workaround

When a PVC is created, if the PVC is in the **Pending** state, you need to take different measures according to the following causes.

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to view details about the PVC.

```
# kubectl describe pvc mypvc
```

Step 3 Perform the corresponding operation according to the **Events** information in the detailed PVC information.

- If the PVC is in the **Pending** state due to cause 1, perform the following steps.

```
Events:
  Type     Reason              Age                From                         Message
  ----     ------              ----               ----                         -------
  Warning  ProvisioningFailed  0s (x15 over 3m24s)  persistentvolume-controller
storageclass.storage.k8s.io "mysc" not found
```


a. Delete the PVC. For details, see **6.2.5 Deleting a PVC**.
b. Create a StorageClass. For details, see **6.1.1 Creating a StorageClass**.
c. Create a PVC. For details, see **6.2.1 Creating a PVC**.

- If the PVC is in the **Pending** state due to cause 2, perform the following steps.

```
Events:
  Type     Reason              Age
From                                                                                                    Message
  ----     ------              ----             ----
-------
  Normal   Provisioning        63s (x3 over 64s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  External provisioner is provisioning volume for claim "default/mypvc"
  Warning  ProvisioningFailed  63s (x3 over 64s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = failed to select pool, the capability filter failed, error: failed to select pool, the final filter field: replication, parameters map[allocType:thin replication:True size: 1099511627776 volumeType:lun]. please check your storage class
```

a. Delete the PVC. For details, see **6.2.5 Deleting a PVC**.
b. Delete the StorageClass. For details, see **6.1.2 Deleting a StorageClass**.
c. Modify the **StorageClass.yaml** file based on the **Events** information.
d. Create a StorageClass. For details, see **6.1.1 Creating a StorageClass**.
e. Create a PVC. For details, see **6.2.1 Creating a PVC**.

- If the PVC is in the **Pending** state due to cause 3, contact Huawei engineers.

```
Events:
  Type     Reason              Age
From                                                                                                    Message
  ----     ------              ----             ----
-------
  Normal   Provisioning        63s (x4 over 68s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  External provisioner is provisioning volume for claim "default/mypvc"
  Warning  ProvisioningFailed  62s (x4 over 68s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = Create volume map[ALLOCTYPE:1 CAPACITY:20 DESCRIPTION:Created from Kubernetes CSI NAME:pvc-63ebfda5-4cf0-458e-83bd-ecc PARENTID:0] error: 50331651
```

- If the PVC is in the **Pending** state due to cause 4, perform the following steps.

```
Events:
  Type     Reason              Age
From                                                                                                    Message
  ----     ------              ----             ----
-------
  Normal   Provisioning        63s (x3 over 52s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  External provisioner is provisioning volume for claim "default/mypvc"
  Warning  ProvisioningFailed  63s (x3 over 52s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

a. Wait for 10 minutes and check the PVC details again by referring to this section.
b. If it is still in the **Pending** state, contact Huawei engineers.

- If the PVC is in the **Pending** state due to cause 5, contact Huawei engineers.

----End

9.7 Before a PVC Is Deleted, the PVC Is in the Pending State

Symptom

Before a PVC is deleted, the PVC is in the **Pending** state.

Environment Configuration

Kubernetes version: 1.13 or later

Root Cause Analysis

Cause 1: A StorageClass with the specified name is not created in advance. As a result, Kubernetes cannot find the specified StorageClass name when a PVC is created.

Cause 2: The storage pool capability does not match the StorageClass capability. As a result, huawei-csi fails to select a storage pool.

Cause 3: An error code (for example, 50331651) is returned by a RESTful interface of the storage. As a result, huawei-csi fails to create a PVC.

Cause 4: The storage does not return a response within the timeout period set by huawei-csi. As a result, huawei-csi returns a timeout error to Kubernetes.

Cause 5: Other causes.

Solution or Workaround

To delete a PVC in the **Pending** state, you need to take different measures according to the following causes.

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to view details about the PVC.

```
# kubectl describe pvc mypvc
```

Step 3 Perform the corresponding operation according to the **Events** information in the detailed PVC information.

- If the PVC is in the **Pending** state due to cause 1, run the **kubectl delete pvc mypvc** command to delete the PVC.

```
Events:
  Type     Reason              Age                From                         Message
  ----     ------              ----               ----                         -------
  Warning  ProvisioningFailed  0s (x15 over 3m24s)  persistentvolume-controller
storageclass.storage.k8s.io "mysc" not found
```

- If the PVC is in the **Pending** state due to cause 2, run the **kubectl delete pvc mypvc** command to delete the PVC.

```
Events:
  Type     Reason              Age
From                                                         Message
  ----     ------              ----               ----
```



```
-------
  Normal   Provisioning          63s (x3 over 64s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  External provisioner is provisioning volume for claim "default/mypvc"
  Warning  ProvisioningFailed    63s (x3 over 64s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = failed to select pool, the capability filter failed, error: failed to select pool, the final filter field: replication, parameters map[allocType:thin replication:True size:1099511627776 volumeType:lun]. please check your storage class
```

- If the PVC is in the **Pending** state due to cause 3, run the **kubectl delete pvc mypvc** command to delete the PVC.

```
Events:
  Type     Reason              Age                From                                                                                                  Message
  ----     ------              ----               ----
-------
  Normal   Provisioning        63s (x4 over 68s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  External provisioner is provisioning volume for claim "default/mypvc"
  Warning  ProvisioningFailed  62s (x4 over 68s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = Create volume map[ALLOCTYPE:1 CAPACITY:20 DESCRIPTION:Created from Kubernetes CSI NAME:pvc-63ebfda5-4cf0-458e-83bd-ecc PARENTID:0] error: 50331651
```

- If the PVC is in the **Pending** state due to cause 4, contact Huawei engineers.

```
Events:
  Type     Reason              Age                From                                                                                                  Message
  ----     ------              ----               ----
-------
  Normal   Provisioning        63s (x3 over 52s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  External provisioner is provisioning volume for claim "default/mypvc"
  Warning  ProvisioningFailed  63s (x3 over 52s)  csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515  failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

- If the PVC is in the **Pending** state due to cause 5, contact Huawei engineers.

----End

9.8 When a Pod Is Created, the Pod Is in the ContainerCreating State

Symptom

A Pod is created. After a period of time, the Pod is still in the **ContainerCreating** state. Check the log information (for details, see **9.1 Viewing Log Information**). The error message "Fibre Channel volume device not found" is displayed.

Environment Configuration

Kubernetes version: 1.13 or later

Root Cause Analysis

This problem occurs because residual disks exist on the host node. As a result, disks fail to be found when a Pod is created next time.

Solution or Workaround

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to query information about the node where the Pod resides.

```
# kubectl get pod -o wide
NAME      READY   STATUS              RESTARTS   AGE     IP            NODE    NOMINATED NODE
READINESS GATES
mypod     0/1     ContainerCreating   0          51s     10.244.1.224  node1   <none>          <none>
```

Step 3 Delete the Pod. For details, see **6.3.2 Deleting a Pod**.

Step 4 Use a remote access tool, such as PuTTY, to log in to the *node1* node in the Kubernetes cluster through the management IP address. *node1* indicates the node queried in **Step 2**.

Step 5 Clear the residual drive letters. For details, see **Solution or Workaround**.

----End

10 Appendix

10.1 Example ALUA Configuration Policy of OceanStor V3/V5 and OceanStor Dorado V3

Example 1: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
            "*": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0", "PATHTYPE": "0"},
            "node1": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0", "PATHTYPE":
"1"}}}
...
```

If the host name is **node1**, both of the preceding ALUA configuration sections can be used to configure initiators. According to the configuration policy rules in **7.4.1 Configuring ALUA for OceanStor V3/V5 and OceanStor Dorado V3**, the priority of the second configuration section (where **HostName** is **node1**) is higher than that of the first configuration section (where **HostName** is *****).

Example 2: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
            "node[0-9]": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0",
"PATHTYPE": "0"},
            "node[5-7]": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0",
"PATHTYPE": "1"}}}
...
```

If the host name is **node6**, both of the preceding ALUA configuration sections can be used to configure initiators. According to the configuration policy rules in **7.4.1 Configuring ALUA for OceanStor V3/V5 and OceanStor Dorado V3**, select the first ALUA configuration section to configure initiators.

Example 3: The configuration file content is as follows:


```
...
"parameters": {..., "ALUA": {
        "^node1$": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0",
"PATHTYPE": "0"},
        "^node10$": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0",
"PATHTYPE": "1"}}}
...
```

According to the configuration policy rules in **7.4.1 Configuring ALUA for OceanStor V3/V5 and OceanStor Dorado V3**: For host **node1**, select the first ALUA configuration section to configure initiators. For host **node10**, select the second ALUA configuration section to configure initiators. **^** matches the beginning of a character string, and **$** matches the end of a character string.

10.2 Example ALUA Configuration Policy of OceanStor Dorado V6

Example 1: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
        "*": {"accessMode": "1", "hyperMetroPathOptimized": "1"},
        "node1": {"accessMode": "1", "hyperMetroPathOptimized": "0"}}}
...
```

If the host name is **node1**, both of the preceding ALUA configuration sections can be used to configure initiators. According to the configuration policy rules in **7.4.2 Configuring ALUA for OceanStor Dorado V6**, the priority of the second configuration section (where **HostName** is **node1**) is higher than that of the first configuration section (where **HostName** is *****).

Example 2: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
        "node[0-9]": {"accessMode": "1", "hyperMetroPathOptimized": "1"},
        "node[5-7]": {"accessMode": "1", "hyperMetroPathOptimized": "0"}}}
...
```

If the host name is **node6**, both of the preceding ALUA configuration sections can be used to configure initiators. According to the configuration policy rules in **7.4.2 Configuring ALUA for OceanStor Dorado V6**, select the first ALUA configuration section to configure initiators.

Example 3: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
        "^node1$": {"accessMode": "1", "hyperMetroPathOptimized": "1"},
        "^node10$": {"accessMode": "1", "hyperMetroPathOptimized": "0"}}}
...
```

According to the configuration policy rules in **7.4.2 Configuring ALUA for OceanStor Dorado V6**: For host **node1**, select the first ALUA configuration section to configure initiators. For host **node10**, select the second ALUA configuration section to configure initiators. **^** matches the beginning of a character string, and **$** matches the end of a character string.

10.3 Example ALUA Configuration Policy of Distributed Storage

Example 1: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
            "*": {"switchoverMode": "Enable_alua", "pathType": "optimal_path"},
            "node1": {"switchoverMode": "Enable_alua", "pathType": "non_optimal_path"}}}
...
```

If the host name is **node1**, both of the preceding ALUA configuration sections can be used to configure initiators. According to the configuration policy rules in **7.4.3 Configuring ALUA for Distributed Storage**, the priority of the second configuration section (where **HostName** is **node1**) is higher than that of the first configuration section (where **HostName** is *****).

Example 2: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
            "node[0-9]": {"switchoverMode": "Enable_alua", "pathType": "optimal_path"},
            "node[5-7]": {"switchoverMode": "Enable_alua", "pathType": "non_optimal_path"}}}
...
```

If the host name is **node6**, both of the preceding ALUA configuration sections can be used to configure initiators. According to the configuration policy rules in **7.4.3 Configuring ALUA for Distributed Storage**, select the first ALUA configuration section to configure initiators.

Example 3: The configuration file content is as follows:

```
...
"parameters": {..., "ALUA": {
            "^node1$": {"switchoverMode": "Enable_alua", "pathType": "optimal_path"},
            "^node10$": {"switchoverMode": "Enable_alua", "pathType": "non_optimal_path"}}}
...
```

According to the configuration policy rules in **7.4.3 Configuring ALUA for Distributed Storage**: For host **node1**, select the first ALUA configuration section to configure initiators. For host **node10**, select the second ALUA configuration section to configure initiators. **^** matches the beginning of a character string, and **$** matches the end of a character string.

