eSDK Huawei Storage Kubernetes CSI Plugins V3.1.0

User Guide

Issue 03

Date 2022-12-09

HUAWEI TECHNOLOGIES CO., LTD.

Copyright © Huawei Technologies Co., Ltd. 2022. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei 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://e.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 hazard with a medium level of risk which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazard with a low level of risk which, if not avoided, could 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.

Copyright © Huawei Technologies Co., Ltd.

Change History

Issue	Date	Description
03	2022-12-09	This issue is the third official release.
02	2022-10-19	This issue is the second official release.
01	2022-09-30	This issue is the first official release.

Copyright © Huawei Technologies Co., Ltd.

Contents

About This Document ii

1 Overview 1

2 Compatibility and Features 3

2.1 Kubernetes and OS Compatibility 3

2.2 Compatibility with Huawei Enterprise Storage 4

2.3 Compatibility with Huawei Distributed Storage 6

3 Installation Preparations 9

3.1 Prerequisites 9

3.2 Downloading Huawei CSI Software Package 10

3.3 Uploading a Huawei CSI Image 11

3.3.1 Uploading an Image to the Image Repository 11

3.3.2 Importing an Image to All Nodes 11

3.4 Checking the Images on Which CSI Depends 12

3.5 Checking Volume Snapshot-Dependent Components 14

3.6 Checking the Host Multipathing Configuration 15

3.7 Checking the Accounts on Huawei Storage 16

3.8 Checking the Status of Host-Dependent Software 17

3.9 Communication Matrix 17

4 Installing Huawei CSI 19

4.1 Installing Huawei CSI Using Helm 19

4.1.1 Preparing the values.yaml File 20

4.1.2 Installing Huawei CSI 38

4.2 Manually Compiling ConfigMap Files to Install Huawei CSI 41

4.2.1 Creating ConfigMap Files Required for Running Huawei CSI 41

4.2.1.1 Connecting to Enterprise Storage SAN over iSCSI 41

4.2.1.2 Connecting to Enterprise Storage SAN over FC 44

4.2.1.3 Connecting to Enterprise Storage NAS over NFS 48

4.2.1.4 Connecting to Enterprise Storage SAN over NVMe over RoCE 51

4.2.1.5 Connecting to Enterprise Storage SAN over NVMe over FC 54

4.2.1.6 Connecting to Distributed Storage SAN over SCSI 57

4.2.1.7 Connecting to Distributed Storage SAN over iSCSI 60

4.2.1.8 Connecting to Distributed Storage NAS over NFS 63

 Copyright © Huawei Technologies Co., Ltd.

4.2.1.9 Connecting to Distributed Storage NAS over DPC 66
4.2.1.10 Connecting to Huawei Storage Using vStores or Accounts 68
4.2.1.11 Provisioning HyperMetro Volumes at Backends Using CSI 69
4.2.1.12 Connecting to Multiple Backends Using CSI 71
4.2.2 Starting huawei-csi Services 72
5 Uninstalling Huawei CSI 77
5.1 Uninstalling huawei-csi Using Helm 77
5.2 Manually Uninstalling huawei-csi 78
5.2.1 Uninstalling the huawei-csi-node Service 78
5.2.2 Uninstalling the huawei-csi-controller Service 79
5.2.3 Deleting the huawei-csi-configmap Object 79
5.2.4 Deleting the huawei-csi-secret Object 79
5.2.5 Deleting the RBAC Permission 80
5.2.6 Deleting the Image of the Earlier Version 81
5.3 Uninstalling the Snapshot-Dependent Component Service 82
6 Upgrade/Rollback Operations 83
6.1 Upgrading or Rolling Back Huawei CSI Using Helm 83
6.1.1 Upgrading Huawei CSI 83
6.1.2 Rolling Back CSI 84
6.2 Manually Compiling ConfigMap Files to Upgrade or Roll Back Huawei CSI 85
6.2.1 Uninstalling Original CSI 85
6.2.2 Installing New CSI 85
7 Using Huawei CSI 87
7.1 Managing a PV/PVC 87
7.1.1 Creating a PVC 87
7.1.1.1 Dynamic Volume Provisioning 88
7.1.1.1.1 Configuring a StorageClass 88
7.1.1.1.2 Configuring a PVC 103
7.1.1.2 Static Volume Provisioning 106
7.1.1.2.1 Configuring a PV 106
7.1.1.2.2 Configuring a PVC 112
7.1.2 Expanding the Capacity of a PVC 115
7.1.3 Cloning a PVC 116
7.1.4 Creating a PVC Using a Snapshot 117
7.2 Creating a VolumeSnapshot 118
7.2.1 Configuring a VolumeSnapshotClass 118
7.2.2 Configuring a VolumeSnapshot 119
8 Advanced Features 121
8.1 Configuring ALUA 121
8.1.1 Configuring ALUA Using Helm 121
8.1.1.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend 121

 Copyright © Huawei Technologies Co., Ltd.

8.1.1.2 Configuring ALUA Parameters for a Distributed Storage Backend 125
8.1.2 Manually Configuring ALUA 127
8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend 127
8.1.2.2 Configuring ALUA Parameters for a Distributed Storage Backend 131
8.2 Configuring Storage Topology Awareness 133
8.2.1 Configuring Storage Topology Awareness Using Helm 134
8.2.2 Manually Configuring Storage Topology Awareness 137
9 Common Operations 140
9.1 Updating the User Name or Password of a Storage Device Configured on CSI 140
9.2 Updating the configmap Object of huawei-csi 141
9.2.1 Updating the configmap Object Using Helm 141
9.2.2 Manually Updating the configmap Object 142
9.3 Adding a Backend for huawei-csi 145
9.3.1 Adding a Backend Using Helm 146
9.3.2 Manually Adding a Backend 147
9.4 Updating the huawei-csi-controller Service 147
9.4.1 Updating the controller Service Using Helm 147
9.4.2 Manually Updating the controller Service 148
9.5 Updating the huawei-csi-node Service 149
9.5.1 Updating the node Service Using Helm 149
9.5.2 Manually Updating the node Service 150
9.6 Modifying the Log Output Mode 150
9.6.1 Modifying the Log Output Mode of the controller or node Service Using Helm 151
9.6.2 Manually Modifying the Log Output Mode of the huawei-csi-controller Service 151
9.6.3 Manually Modifying the Log Output Mode of the huawei-csi-node Service 153
9.7 Enabling the ReadWriteOncePod Feature Gate 154
9.8 Configuring Access to the Kubernetes Cluster as a Non-root User 156
10 FAQ 157
10.1 How Do I View Huawei CSI Logs? 158
10.2 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 159
10.3 Failed to Start the huawei-csi-node Service with Error Message "/var/lib/iscsi is not a directory" Reported 159
10.4 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 160
10.5 Failed to Start huawei-csi Services with the Status Displayed as InvalidImageName 163
10.6 When a PVC Is Created, the PVC Is in the Pending State 164
10.7 Before a PVC Is Deleted, the PVC Is in the Pending State 166
10.8 When a Pod Is Created, the Pod Is in the ContainerCreating State 167
10.9 A Pod Is in the ContainerCreating State for a Long Time When It Is Being Created 168
10.10 A Pod Fails to Be Created and the Log Shows That the Execution of the mount Command Times Out 168
10.11 A Pod Fails to Be Created and the Log Shows That the mount Command Fails to Be Executed 169

 Copyright © Huawei Technologies Co., Ltd.

10.12 How Do I Download a Container Image to the Local PC?....169
10.13 How Do I Obtain CSI Version Information?....170
10.14 Common Problems and Solutions for Interconnecting with the Tanzu Kubernetes Cluster....171
10.14.1 A Pod Cannot Be Created Because the PSP Permission Is Not Created....171
10.14.2 Changing the Mount Point of a Host....172
10.14.3 Changing the Default Port of the livenessprobe Container....173
10.15 Common Problems and Solutions for Using the Tanzu Kubernetes Cluster....173
10.15.1 Failed to Create an Ephemeral Volume....173
10.16 Failed to Expand the Capacity of a Generic Ephemeral Volume....174
10.17 Failed to Expand the PVC Capacity Because the Target Capacity Exceeds the Storage Pool Capacity....175

11 Appendix....176

11.1 Example ALUA Configuration Policy of OceanStor V3/V5 and OceanStor Dorado V3....176
11.2 Example ALUA Configuration Policy of OceanStor Dorado 6.*x*....177
11.3 Example ALUA Configuration Policy of Distributed Storage....178
11.4 Installing Helm 3....178
11.5 Creating a CSI Image....179

 Copyright © Huawei Technologies Co., Ltd.

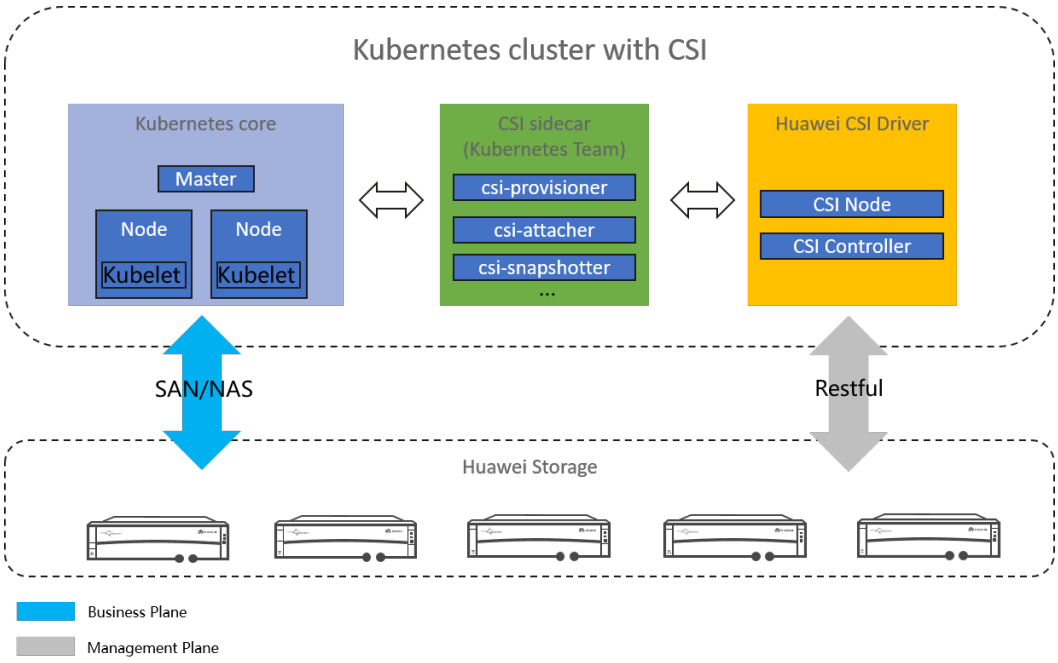
1 Overview

Kubernetes (K8s or "kube" for short) is a portable, extensible, and open-source platform for managing containerized workloads and services.

Container Storage Interface (CSI) is an industry standard used to expose block and file storage systems to container workloads on container orchestration systems (COs) such as Kubernetes. Huawei CSI plug-in is used to communicate with Huawei enterprise storage and distributed storage products and provide storage services for Kubernetes container workloads. It is a mandatory plug-in used by Huawei enterprise storage and distributed storage in the Kubernetes environment.

The following figure shows the overall structure of Kubernetes, Huawei CSI, and Huawei storage.

Figure 1-1 CSI overall architecture



- Kubernetes uses a series of officially maintained sidecar components to register and listen to Kubernetes object resources and invoke CSI Driver when necessary.

 Copyright © Huawei Technologies Co., Ltd.

- Huawei CSI Driver implements the invocation initiated by sidecar on Huawei storage. For example, the operation of creating a **PersistentVolume (PV)** is implemented as follows: creating a LUN or file system on Huawei storage.
  - CSI Controller: a Pod that runs independently in Deployment mode. It is used to interact with storage and create and delete resources on the storage side, such as creating a LUN or file system and expanding capacity.
  - CSI Node: a Pod that runs on Kubernetes worker nodes in DaemonSet mode. It is used to mount and unmount a LUN/file system provided by Huawei storage on worker nodes, and format a LUN into a local file system.
- Huawei storage provides SAN/NAS storage resource services for Kubernetes worker nodes through multiple protocols. Huawei storage communicates with Huawei CSI drivers through RESTful.

This document describes how to install, deploy, and use the Huawei CSI V3.1.0 plug-in.

Copyright © Huawei Technologies Co., Ltd.

2 Compatibility and Features

This chapter describes the container management platforms, operating systems (OSs), and multipathing software supported by Huawei CSI plug-in, as well as the features and functions provided by the CSI plug-in when working with Huawei storage.

2.1 Kubernetes and OS Compatibility

2.2 Compatibility with Huawei Enterprise Storage

2.3 Compatibility with Huawei Distributed Storage

2.1 Kubernetes and OS Compatibility

Huawei CSI plug-in supports the following container management platforms.

Table 2-1 Supported container management platforms

Container Management Platform	Version
Kubernetes	1.19, 1.20, 1.21, 1.22, 1.23, 1.24
Red Hat OpenShift	4.6 EUS, 4.7, 4.8, 4.9, 4.10
Tanzu Kubernetes	TKGI 1.14.1 + Kubernetes 1.23.7

NOTICE

- For details about how to connect Huawei CSI to Red Hat OpenShift, see ***eSDK Enterprise Storage Plugins User Guide (Kubernetes CSI for Red Hat OpenShift)***.
- For FAQs about connecting Huawei CSI to Tanzu Kubernetes, see **10.14 Common Problems and Solutions for Interconnecting with the Tanzu Kubernetes Cluster**.

Table 2-2 lists the OSs and multipathing software supported by the Huawei CSI plug-in.

 Copyright © Huawei Technologies Co., Ltd.

Table 2-2 Supported OSs and multipathing software

OS Name	OS Version	Native DM-Multipath Version	Huawei UltraPath version
CentOS x86_64	7.6, 7.7, 7.9	Delivered with the OS, supporting FC/iSCSI	UltraPath 31.1.0, supporting FC/iSCSI
CentOS x86_64	8.2	Delivered with the OS, supporting FC/iSCSI	UltraPath 31.1.0, supporting FC/iSCSI UltraPath-NVMe 31.1.RC8, supporting NVMe over RoCE/NVMe over FC
SUSE 15 x86_64	SP2	Delivered with the OS, supporting FC/iSCSI	UltraPath 31.1.0, supporting FC/iSCSI UltraPath-NVMe 31.1.RC8, supporting NVMe over RoCE/NVMe over FC
Red Hat CoreOS x86_64	4.6, 4.7, 4.8, 4.9, 4.10	Delivered with the OS, supporting FC/iSCSI	Not supported
Ubuntu x86_64	18.04, 20.04	Delivered with the OS, supporting FC/iSCSI	Not supported
Kylin V10 x86_64	SP1, SP2	Delivered with the OS, supporting FC/iSCSI	Not supported
Kylin V10 ARM	SP1, SP2	Delivered with the OS, supporting FC/iSCSI	Not supported
Debian x86_64	11	Delivered with the OS, supporting FC/iSCSI	Not supported

2.2 Compatibility with Huawei Enterprise Storage

Huawei CSI plug-in is compatible with Huawei OceanStor series all-flash storage and hybrid flash storage. The following table lists the supported storage versions.

Table 2-3 Supported Huawei enterprise storage

Storage Product	Version
OceanStor F V3/V3	V300R006
OceanStor F V5/V5	V500R007, V500R007 Kunpeng

Copyright © Huawei Technologies Co., Ltd.

Storage Product	Version
OceanStor Dorado V3	V300R002
OceanStor V6	6.1.3, 6.1.5
OceanStor Dorado V6	6.0.0, 6.0.1, 6.1.0, 6.1.2, 6.1.3, 6.1.5

Huawei CSI plug-in supports the following features for Huawei enterprise storage.

Table 2-4 Features supported by Huawei enterprise storage and constraints

Feature	OceanStor F V3/V3	OceanStor F V5/V5	OceanStor Dorado V3	OceanStor V6	OceanStor Dorado V6
Static Provisioning	SAN: FC/ iSCSI NAS: NFS 3	SAN: FC/ iSCSI NAS: NFS 3	SAN: FC/ iSCSI	SAN: FC/ iSCSI/NVMe over RoCE/ NVMe over FC NAS: NFS 3/4.0/4.1	SAN: FC/ iSCSI/NVMe over RoCE/ NVMe over FC[1] NAS: NFS 3/4.0/4.1[2]
Dynamic Provisioning					
Expand Persistent Volume	Only volumes created in Dynamic Provisioning mode are supported.				
Create VolumeSnapshot	Only non-HyperMetro volumes created in Dynamic Provisioning mode are supported.				
Delete VolumeSnapshot	Supported	Supported	Supported	Supported	Supported
Restore VolumeSnapshot	Supported	Supported	Supported	SAN: supported NAS: supported only in 6.1.5	SAN: supported NAS: supported only in 6.1.5
Clone Persistent Volume	Only volumes created in Dynamic Provisioning mode are supported.			SAN: supports volumes created in Dynamic Provisioning mode. NAS: Only 6.1.5 supports volumes created in Dynamic Provisioning mode.	

 Copyright © Huawei Technologies Co., Ltd.

Feature	OceanStor F V3/V3	OceanStor F V5/V5	OceanStor Dorado V3	OceanStor V6	OceanStor Dorado V6
Raw Block Volume	Only SAN volumes are supported.	Only SAN volumes are supported.	Only SAN volumes are supported.	Only SAN volumes are supported.	Only SAN volumes are supported.
Topology	Supported	Supported	Supported	Supported	Supported
Generic Ephemeral Volumes	Supported	Supported	Supported	Supported	Supported
Access Mode	RWO/ROX/RWOP: supported by all types of volumes. RWOP is supported only by Kubernetes 1.22 and later versions. RWX: supported only by Raw Block volumes and NFS volumes.				
QoS	Supported	Supported	Supported	Supported	Supported
Application type	N/A	N/A	N/A	Supported	Supported
Volume HyperMetro	Not supported	Not supported	N/A	Only NAS volumes are supported.	
Storage multi-tenant	Only NAS volumes are supported.		N/A	Only NAS volumes are supported.[3]	

- Note 1: If NVMe over RoCE/NVMe over FC is used, the nvme-cli tool version of the compute nodes must be 1.9 or later.
- Note 2: Only OceanStor Dorado V6 6.1.0 and later versions support NFS. Only OceanStor Dorado V6 6.1.3 and later versions support NFS 4.1.
- Note 3: Only OceanStor Dorado V6 6.1.3 and later versions support multi-tenant.

2.3 Compatibility with Huawei Distributed Storage

Huawei CSI plug-in is compatible with Huawei OceanStor series distributed storage systems. The following table lists the supported storage versions.

 Copyright © Huawei Technologies Co., Ltd.

Table 2-5 Supported Huawei distributed storage

Storage Product	Version
FusionStorage	V100R006C30
FusionStorage Block	8.0.0, 8.0.1
OceanStor Pacific series	8.1.0, 8.1.1, 8.1.2, 8.1.3

Huawei CSI plug-in supports the following features for Huawei distributed storage.

Table 2-6 Features supported by Huawei distributed storage and constraints

Feature	FusionStorage	FusionStorage Block	OceanStor Pacific Series
Static Provisioning	SAN: iSCSI/SCSI	SAN: iSCSI/SCSI	SAN: iSCSI/SCSI NAS: DPC[1]/NFS 3/4.1[2]
Dynamic Provisioning			
Expand Persistent Volume	Only SAN volumes created in Dynamic Provisioning mode are supported.		
Create VolumeSnapshot	Only SAN volumes created in Dynamic Provisioning mode are supported.		
Delete VolumeSnapshot	Supported	Supported	Only SAN volume snapshots are supported.
Restore VolumeSnapshot	Supported	Supported	Only SAN volume snapshots are supported.
Clone Persistent Volume	Only SAN volumes created in Dynamic Provisioning mode are supported.		
Raw Block Volume	Only SAN volumes are supported.	Only SAN volumes are supported.	Only SAN volumes are supported.
Topology	Supported	Supported	Supported
Generic Ephemeral Volumes	Supported	Supported	Supported
Access Mode	RWO/ROX/RWOP: supported by all types of volumes. RWOP is supported only by Kubernetes 1.22 and later versions. RWX: supported only by Raw Block volumes and NFS volumes.		

 Copyright © Huawei Technologies Co., Ltd.

Feature	FusionStorage	FusionStorage Block	OceanStor Pacific Series
QoS	Supported	Supported	Only SAN volumes are supported.
Soft quota	Not supported	Not supported	Only NAS volumes are supported.
Storage multi-tenant	Not supported	Not supported	Only NAS volumes are supported.

- Note 1: Only OceanStor Pacific series 8.1.2 and later versions support DPC. For details about whether the **OSs supported by Huawei CSI** support DPC, see the compatibility document of the corresponding product version.
- Note 2: Only OceanStor Pacific series 8.1.2 and later versions support NFS 4.1.

Copyright © Huawei Technologies Co., Ltd.

3 Installation Preparations

Before installing Huawei CSI plug-in on your container management platform, you need to create a CSI image, upload or import the image, and configure the host multipathing environment. This chapter describes the preparations for the installation.

3.1 Prerequisites

3.2 Downloading Huawei CSI Software Package

3.3 Uploading a Huawei CSI Image

3.4 Checking the Images on Which CSI Depends

3.5 Checking Volume Snapshot-Dependent Components

3.6 Checking the Host Multipathing Configuration

3.7 Checking the Accounts on Huawei Storage

3.8 Checking the Status of Host-Dependent Software

3.9 Communication Matrix

3.1 Prerequisites

Before performing the operations described in this chapter, ensure that the following conditions are met:

- A container management platform has been deployed, is running properly, and meets the requirements described in **2.1 Kubernetes and OS Compatibility**.
- Initial configuration for interconnecting with Huawei enterprise storage has been completed, including storage pool division and port configuration. The version of the storage product meets the requirements in **2.2 Compatibility with Huawei Enterprise Storage**.
- Initial configuration for interconnecting with Huawei distributed storage has been completed, including storage pool division and port configuration. The version of the storage product meets the requirements in **2.3 Compatibility with Huawei Distributed Storage**.

 Copyright © Huawei Technologies Co., Ltd.

- The connectivity between Huawei storage and the container platform host has been configured. Based on your plan, ensure that the software clients required by the corresponding protocol, such as the iSCSI client and NFS client, have been installed on the worker nodes in the container cluster.
- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes. For details, see **Table 2-2**.
- All worker nodes of Kubernetes communicate properly with the service IP address of the storage device to be connected. In iSCSI scenarios, the **ping** command can be used to verify the connectivity.
- A Linux host with Docker installed is available, and the host can access the Huawei Kubernetes CSI warehouse.

3.2 Downloading Huawei CSI Software Package

This section describes how to download the software package and the component structure of the software package.

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

Step 2 Download the software package of the 3.1.0 version based on the storage type and CPU architecture.

NOTE

Software package naming rule: Storage type + Plug-in name (**Kubernetes_CSI_Plugin**) + Version number + CPU architecture

For example, if distributed storage is used to connect to x86 hosts, the name of the software package to be downloaded is **eSDK_Huawei_Storage_Kubernetes_CSI_Plugin_V3.1.0_X86_64.zip**.

Step 3 Decompress the downloaded software package. **Table 3-1** shows the component structure of the software package.

Table 3-1 Component description

Component	Description
image/huawei-csi-v3.1.0-*arch*.tar	huawei-csi image. *arch* is **x86** or **arm**.
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.
helm	Helm component used to deploy Huawei CSI.
deploy	.yaml sample file used during CSI deployment.
examples	.yaml sample file used during CSI use.

Copyright © Huawei Technologies Co., Ltd.

Component	Description
tools	Script used to upload images when no image repository is available.

----End

3.3 Uploading a Huawei CSI Image

Currently, Huawei has provided the **huawei-csi** image for users. For details about how to obtain the image file, see **Table 3-1**. If you need to create an image again, see **11.5 Creating a CSI Image**.

To use the CSI image on the container management platform, you need to import the CSI image to the cluster in advance using either of the following methods:

- (Recommended) Use Docker to upload the CSI image to the image repository.
- Use the image upload script to import the CSI image to all nodes where Huawei CSI needs to be deployed.

3.3.1 Uploading an Image to the Image Repository

Prerequisites

A Linux host with Docker installed is available, and the host can access the image repository.

Procedure

Step 1 Run the **docker load -i huawei-csi.tar** command to import the CSI image to the current node.

```
# docker load -i huawei-csi.tar
Loaded image: huawei-csi:3.1.0
```

Step 2 Run the **docker tag huawei-csi:3.1.0 repo.huawei.com/huawei-csi:3.1.0** command to add the image repository address to the image tag. **repo.huawei.com** indicates the image repository address.

```
# docker tag huawei-csi:3.1.0 repo.huawei.com/huawei-csi:3.1.0
```

Step 3 Run the **docker push repo.huawei.com/huawei-csi:3.1.0** command to upload the CSI image to the image repository. **repo.huawei.com** indicates the image repository address.

```
# docker push repo.huawei.com/huawei-csi:3.1.0
```

----End

3.3.2 Importing an Image to All Nodes

If the image has been uploaded to the image repository, skip this section.

Copyright © Huawei Technologies Co., Ltd.

Prerequisites

- The host where the image is located can communicate with all hosts to which the image is to be imported using SSH.
- The **expect**, **sshpass**, and **scp** software packages have been installed on the host where the image is located.

Procedure

Step 1 Run the **vi** *worker-list.txt* command to create the **worker-list.txt** configuration file.

```
# vi worker-list.txt
```

Step 2 Configure the **worker-list.txt** file. The template of the **worker-list.txt** file is as follows. Press **I** or **Insert** to enter the editing mode and add node information. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
# ip
192.168.128.16
192.168.128.17
```

Step 3 Upload and import an image.

- If containerd is used as the container runtime, run the **./containerd-upload.sh** *worker-list.txt huawei-csi.tar* command, enter the user name and password as prompted, and upload and import an image.

  ```
  # ./containerd-upload.sh worker-list.txt huawei-csi.tar
  ```

- If Docker is used as the container runtime, run the **./docker-upload.sh** *worker-list.txt huawei-csi.tar* command, enter the user name and password as prompted, and upload and import an image.

  ```
  # ./docker-upload.sh worker-list.txt huawei-csi.tar
  ```

Step 4 Check whether the image is successfully imported.

1. If **All images are uploaded successfully** is displayed at the end of the script, the image has been successfully imported to all nodes.

   ```
   All images are uploaded successfully
   ```

2. If the following information is displayed at the end of the script, the image fails to be imported to the nodes in the list. In this case, check the logs for the cause.

   ```
   List of nodes to which the image fails to be imported:
     192.168.128.16
     192.168.128.17
   ```

----End

3.4 Checking the Images on Which CSI Depends

The installation of Huawei CSI depends on the images listed in the following table. If all worker nodes in the cluster have been connected to the Internet and can pull images online, skip this section. If nodes in the cluster cannot connect to the Internet, download the corresponding image file based on the Kubernetes version and upload it to the image repository or import it to all worker nodes in the Kubernetes cluster.

The huawei-csi-controller service depends on the following sidecar images: livenessprobe, csi-provisioner, csi-attacher, csi-resizer, csi-snapshotter, snapshot-controller, and huawei-csi-driver.

 Copyright © Huawei Technologies Co., Ltd.

The huawei-csi-node service depends on the following sidecar images: livenessprobe, csi-node-driver-registrar, and huawei-csi-driver.

For details about the functions and details of each image, see the following table.

Table 3-2 Images on which Huawei CSI depends

Container Name	Container Image	K8s Version Requirements	Feature Description	Official Description
livenessprobe	k8s.gcr.io/sig-storage/livenessprobe:v2.5.0	v1.19+	Monitors the health status of CSI and reports it to Kubernetes so that Kubernetes can automatically detect CSI program problems and restart the Pod to rectify the problems.	View details
csi-resizer	k8s.gcr.io/sig-storage/csi-resizer:v1.3.0	v1.19+	Calls CSI to provide more storage space for a PVC when expanding the capacity of the PVC.	View details
csi-node-driver-registrar	k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0	v1.19+	Obtains CSI information and registers a node with kubelet using the plug-in registration mechanism of kubelet so that Kubernetes can detect the connection between the node and Huawei storage.	View details
csi-snapshotter	k8s.gcr.io/sig-storage/csi-snapshotter:v4.2.1	v1.20+	Calls CSI to create or delete a snapshot on the storage system when creating or deleting a VolumeSnapshot.	View details
	k8s.gcr.io/sig-storage/csi-snapshotter:v3.0.3	v1.19		

Copyright © Huawei Technologies Co., Ltd.

Container Name	Container Image	K8s Version Requirements	Feature Description	Official Description
snapshot-controller	k8s.gcr.io/sig-storage/snapshot-controller:v4.2.1	v1.20+	Listens to the VolumeSnapshot and VolumeSnapshotContent objects in the Kubernetes API and triggers csi-snapshotter to create a snapshot on the storage system when creating or deleting a VolumeSnapshot.	View details
	k8s.gcr.io/sig-storage/snapshot-controller:v3.0.3	v1.19		
csi-provisioner	k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0	v1.20+	• Calls the CSI Controller service to create a LUN or file system on the storage system as a PV and bind the PV to a PVC when creating a PVC. • Calls the CSI Controller service to unbind a PV from a PVC and delete the LUN or file system corresponding to the PV when deleting a PVC.	View details
	quay.io/k8scsi/csi-provisioner:v2.2.2	v1.19		
csi-attacher	k8s.gcr.io/sig-storage/csi-attacher:v3.3.0	v1.19+	Calls the CSI Controller service to perform the "Publish/Unpublish Volume" operation when creating or deleting a Pod.	View details

NOTE

For details about how to download container images to the local host, see **10.12 How Do I Download a Container Image to the Local PC?**.

3.5 Checking Volume Snapshot-Dependent Components

If you need to use volume snapshots and features associated with volume snapshots in the container environment, perform the following steps to check

 Copyright © Huawei Technologies Co., Ltd.

whether volume snapshot-dependent components have been deployed in your environment. If Helm is used to install Huawei CSI, skip this section. If the current Kubernetes version does not support the volume snapshot feature, skip this section.

Procedure

Step 1 Run the following command to view the installation details of snapshot-related resource services.

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

- If the preceding command output is displayed, the snapshot-dependent component services have been installed. In this case, skip this section.
- If any of the services in the preceding command output is not displayed, go to the next step to install it.

Step 2 Copy the **deploy** directory in the Kubernetes CSI component package to any directory. For details about the component package path, see **Table 3-1**.

Step 3 Go to the **deploy** directory and run the following command to install the snapshot-dependent component services.

```
# kubectl apply -f huawei-csi-snapshot-crd.yaml
```

----End

After the installation is complete, you can run the command in **Step 1** to check the installation details of snapshot-related resource services.

3.6 Checking the Host Multipathing Configuration

If you plan to use the FC/iSCSI/NVMe over RoCE/NVMe over FC protocol to access Huawei storage in a container environment, you are advised to use host multipathing software to enhance the link redundancy and performance of the host and storage. If you do not want to use the software, skip this section.

For details about the OSs and multipathing software supported by Huawei CSI, see **Table 2-2**.

NOTE

- If you want to use the FC/iSCSI protocol to connect to Huawei storage, you are advised to use native DM-Multipath provided by the OS.
- If you want to use the NVMe over RoCE/NVMe over FC protocol to connect to Huawei storage, you are advised to use Huawei-developed UltraPath-NVMe.
- If you want to use the SCSI protocol to connect to Huawei storage, disable DM-Multipath provided by the OS.

Prerequisites

Multipathing software has been correctly installed on a host.

- If you use native DM-Multipath provided by the OS, contact your host or OS provider to obtain the documents and software packages required for the installation.

Copyright © Huawei Technologies Co., Ltd.

- If you use Huawei-developed UltraPath or UltraPath-NVMe, contact Huawei engineers to obtain the UltraPath or UltraPath-NVMe documents and software packages. For details about the software package versions, see **Table 2-2**.

Procedure

Step 1 If you use the iSCSI/FC protocol to connect to Huawei enterprise storage, configure and check host multipathing by referring to **Configuring Multipathing** > **Non-HyperMetro Scenarios** in ***OceanStor Dorado 6.x and OceanStor 6.x Host Connectivity Guide for Red Hat***.

Step 2 If you use the NVMe over RoCE/NVMe over FC protocol to connect to Huawei enterprise storage, configure and check host multipathing by referring to **Configuring Multipathing** > **Non-HyperMetro Scenarios** > **UltraPath** in ***OceanStor Dorado 6.x and OceanStor 6.x Host Connectivity Guide for Red Hat***.

Step 3 If you use iSCSI to connect to Huawei distributed storage, configure and check host multipathing by referring to **Configuring Multipathing for an Application Server** in ***FusionStorage 8.0.1 Block Storage Basic Service Configuration Guide***.

Step 4 If you use the native multipathing software provided by the OS, check whether the **/etc/multipath.conf** file contains the following configuration item.

```
defaults {
        user_friendly_names yes
        find_multipaths no
}
```

If the configuration item does not exist, add it to the beginning of the **/etc/multipath.conf** file.

----End

3.7 Checking the Accounts on Huawei Storage

After Huawei storage is connected to the container platform, Huawei CSI needs to manage storage resources on Huawei storage based on service requirements, such as creating and mapping volumes. In this case, Huawei CSI needs to use the accounts created on Huawei storage to communicate with Huawei storage. **Table 3-3** lists the accounts required for different storage devices.

Table 3-3 Account requirements for connecting storage to CSI

Storage Type	User Type	Role	Level	Type
OceanStor V3/V5	System user	Administrator	Administrator	Local user
	vStore user	vStore administrator	Administrator	Local user
OceanStor Dorado V3	System user	Administrator	Administrator	Local user

Copyright © Huawei Technologies Co., Ltd.

Storage Type	User Type	Role	Level	Type
OceanStor 6.1	System user	Administrator	N/A	Local user
OceanStor Dorado 6.1.3/6.1.5	System user	Administrator	N/A	Local user
	vStore user	vStore administrator	N/A	Local user
OceanStor Pacific series	System user	Administrator	N/A	Local user

3.8 Checking the Status of Host-Dependent Software

This section describes how to check whether the status of host-dependent software on worker nodes in a cluster is normal. In this example, the host OS is CentOS 7.9 x86_64.

- Check the status of the iSCSI client.
  ```
  # systemctl status iscsi iscsid
  ```
- Check the status of the NFS client.
  ```
  # systemctl status rpcbind
  ```
- Check the status of DM-Multipath.
  ```
  # systemctl status multipathd.socket multipathd
  ```
- Check the status of UltraPath.
  ```
  # systemctl status nxup
  ```
- Check the status of UltraPath-NVMe.
  ```
  # systemctl status upudev upService_plus
  ```

3.9 Communication Matrix

This section describes the communication relationships among the CSI, storage, and kubelet, covering the ports, protocols, IP addresses, authentication modes, and port descriptions. For details, see **Table 3-4**.

Table 3-4 Communication matrix

Source Device	Host where CSI is located	Host where CSI is located
Source IP Address	IP address of the host where CSI is located	IP address of the host where CSI is located
Source Port	1024 to 65535	1024 to 65535
Destination Device	Storage device	Host where CSI is located
Destination IP Address	Management IP address of the storage device	IP address of the host where CSI is located

Copyright © Huawei Technologies Co., Ltd.

Destination Port	8088	9800/9808
Protocol	HTTPS	HTTP
Authentication Mode	User name and password	None
Port Description	• 1024 to 65535: This port is used to connect CSI to the storage device. The port allocation is determined by the temporary Linux port number range. • 8088: This port is provided by the storage device and is used by CSI to request volume creation, management, and deletion.	• 1024 to 65535: The port allocation is determined by the temporary Linux port number range. • 9800/9808: This port is provided by CSI and used by kubelet to check the CSI health status.

Copyright © Huawei Technologies Co., Ltd.

4 Installing Huawei CSI

This section describes how to install Huawei CSI. You are advised to install Huawei CSI using Helm. The method of manually installing Huawei CSI is retained in the current version but will be deleted in later versions.

Huawei CSI can be installed as the root user or a non-root user. When installing Huawei CSI as a non-root user, ensure that the current user can access the API Server of the Kubernetes cluster. For details about how to configure access to the Kubernetes cluster as a non-root user, see **9.8 Configuring Access to the Kubernetes Cluster as a Non-root User**.

Huawei CSI must be run as the root user.

4.1 Installing Huawei CSI Using Helm

4.2 Manually Compiling ConfigMap Files to Install Huawei CSI

4.1 Installing Huawei CSI Using Helm

This section describes how to install Huawei CSI using Helm 3.

Helm is a software package management tool in the Kubernetes ecosystem. Similar to Ubuntu APT, CentOS YUM, or Python pip, Helm manages Kubernetes application resources. You can use Helm to package, distribute, install, upgrade, and roll back Kubernetes applications in a unified manner.

- For details about how to obtain and install Helm, see **https://helm.sh/docs/intro/install/**.
- For other information about Helm, visit **https://github.com/helm/helm**.

When installing huawei-csi-controller, Helm deploys the following components in the workloads of the Deployment type in the specified namespace:

- huawei-csi-driver: Huawei CSI driver.
- Kubernetes External Provisioner: used to provide volumes.
- Kubernetes External Attacher: used to attach volumes.
- (Optional) Kubernetes External Snapshotter: used to provide snapshot support (installed as CRD).
- (Optional) Kubernetes External Resizer: used to expand the capacity of volumes.

 Copyright © Huawei Technologies Co., Ltd.

When installing huawei-csi-node, Helm deploys the following components in the workloads of the DaemonSet type in the specified namespace:

- huawei-csi-driver: Huawei CSI driver.
- Kubernetes Node Registrar: used to process driver registration.

4.1.1 Preparing the values.yaml File

When using Helm to install CSI, you need to prepare the **values.yaml** file based on the Huawei storage connected during deployment and the features to be used. Huawei CSI provides the **values.yaml** template file in the **helm/esdk** directory of the software package. This section describes the configuration items in the **values.yaml** file and backend configuration examples in typical scenarios.

Configure the following content in the **values.yaml** file:

- **Kubernetes Configuration Items**
- **images Configuration Items**
- **csi_driver Configuration Items**
- **backends Configuration Items**
- **Other Configuration Items**

For details about the **backends** configuration in the following typical scenarios, see the examples.

- **Configuring Multiple Huawei Storage Backends**
- **Configuring Huawei Storage Backends for Multiple vStores or Accounts**
- **Configuring a Storage Backend of the iSCSI Type**
- **Configuring a Storage Backend of the FC Type**
- **Configuring a Storage Backend of the NVMe over RoCE Type**
- **Configuring a Storage Backend of the NVMe over FC Type**
- **Configuring a Storage Backend of the NFS Type**
- **Configuring a Storage Backend of the SCSI Type**
- **Configuring a Storage Backend of the DPC Type**
- **Configuring Storage Backends of the HyperMetro Type**

Kubernetes Configuration Items

The Kubernetes configuration items in the **values.yaml** file are used to configure the context information when Huawei CSI is running. Set the following parameters:

Copyright © Huawei Technologies Co., Ltd.

Table 4-1 Kubernetes configuration items

Parameter	Description	Mandatory	Default Value
kubernetes.namespace	Kubernetes namespace where Huawei CSI is running, which can be customized. The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.	Yes	huawei-csi

NOTICE

It is strongly recommended that a separate namespace be used for Huawei CSI. The **default**, **kube-system**, and **kube-public** namespaces provided by the system are not recommended.

Ensure that the entered namespace exists on Kubernetes. If the namespace does not exist, run the following command to create it. In this example, the namespace for running Huawei CSI is **huawei-csi**.

```
# kubectl create namespace huawei-csi
```

images Configuration Items

The **images** configuration items in the **values.yaml** file are used to configure the component image information on which Huawei CSI depends during running. Set the following parameters:

Table 4-2 images configuration items

Parameter	Description	Mandatory	Default Value
images.sidecar.livenessProbe	**livenessprobe** sidecar image.	Yes	k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
images.sidecar.csiProvisioner	**csi-provisioner** sidecar image.	Yes	k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
images.sidecar.csiAttacher	**csi-attacher** sidecar image.	Yes	k8s.gcr.io/sig-storage/csi-attacher:v3.3.0
images.sidecar.csiResizer	**csi-resizer** sidecar image.	Yes	k8s.gcr.io/sig-storage/csi-resizer:v1.3.0

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Default Value
images.sidecar.csiSnapshotter	**csi-snapshotter** sidecar image.	Yes	k8s.gcr.io/sig-storage/csi-snapshotter:v4.2.1
images.sidecar.snapshotController	**snapshot-controller** sidecar image.	Yes	k8s.gcr.io/sig-storage/snapshot-controller:v4.2.1
images.sidecar.registrar	**csi-node-driver-registrar** sidecar image.	Yes	k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0
images.huaweiCSIService	huawei-csi image.	Yes	-

For details about the value of **huaweiCSIService**, see **3.3 Uploading a Huawei CSI Image**. Use the name and version of the finally generated image. In this example, the huawei-csi image name is **huawei-csi:3.1.0**.

For details about other sidecar image parameters, see **3.4 Checking the Images on Which CSI Depends**. Use the name and version of the finally uploaded image.

csi_driver Configuration Items

The **csi_driver** configuration items include the basic configurations for running Huawei CSI, such as Huawei driver name and multipathing type. Set the following parameters:

Table 4-3 csi_driver configuration items

Parameter	Description	Mandatory	Default Value	Suggestion
csi_driver.driverName	Registered driver name.	Yes	csi.huawei.com	Use the default value.
csi_driver.endpoint	Communication endpoint.	Yes	/csi/csi.sock	Use the default value.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Default Value	Suggestion
csi_driver.connectorThreads	Maximum number of disks that can be concurrently scanned/ detached. The value is an integer ranging from 1 to 10.	Yes	4	A larger value indicates that more concurrent disk scanning and detaching operations are performed on a single node at the same time. When DM-Multipath is used, a large number of concurrent requests may cause unknown problems and affect the overall time.
csi_driver.volumeUseMultipath	Whether to use multipathing software. The value is a Boolean value.	Yes	true	It is strongly recommended that multipathing software be enabled to enhance the redundancy and performance of storage links.
csi_driver.scsiMultipathType	Multipathing software used when the storage protocol is **fc** or **iscsi**. The following parameter values can be configured: • DM-multipath • HW-UltraPath • HW-UltraPath-NVMe	Mandatory when **volumeUseMultipath** is set to **TRUE**.	DM-multipath	The **DM-multipath** value is recommended.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Default Value	Suggestion
csi_driver.nvmeMultipathType	Multipathing software used when the storage protocol is **roce** or **fc-nvme**. Only **HW-UltraPath-NVMe** is supported.	Mandatory when **volumeUseMultipath** is set to **TRUE**.	HW-UltraPath-NVMe	-
csi_driver.scanVolumeTimeout	Timeout interval for waiting for multipathing aggregation when DM-Multipath is used on the host. The value ranges from 1 to 600 seconds.	Yes	3	-
csi_driver.backendUpdateInterval	Interval for updating backend capabilities. The value ranges from 60 to 600 seconds.	Yes	60	-

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Default Value	Suggestion
csi_driver.controllerLogging.module	Record type of the controller log. The following parameter values can be configured: • file • console	Yes	file	When the value is **file**, logs are retained in the specified directory of the node. When the Pod where CSI is located is destroyed, logs are still retained. When the value is **console**, logs are retained in the temporary space of the Pod where CSI is located. When the Pod where CSI is located is destroyed, the logs are also destroyed.
csi_driver.controllerLogging.level	Output level of the controller log. The following parameter values can be configured: • debug • info • warning • error • fatal	Yes	info	-
csi_driver.controllerLogging.fileDir	Directory of the controller log in **file** output mode.	Yes	/var/log/huawei	Ensure that the directory has sufficient space for storing logs. It is recommended that the space be greater than or equal to 200 MB.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Default Value	Suggestion
csi_driver.controllerLogging.fileSize	Size of a single controller log file in **file** output mode.	Yes	20M	-
csi_driver.controllerLogging.maxBackups	Maximum number of controller log file backups in **file** output mode.	Yes	9	-
csi_driver.nodeLogging.module	Record type of the node log. The following parameter values can be configured: ● file ● console	Yes	file	When the value is **file**, logs are retained in the specified directory of the node. When the Pod where CSI is located is destroyed, logs are still retained. When the value is **console**, logs are retained in the temporary space of the Pod where CSI is located. When the Pod where CSI is located is destroyed, the logs are also destroyed.
csi_driver.nodeLogging.level	Output level of the node log. The following parameter values can be configured: ● debug ● info ● warning ● error ● fatal	Yes	info	-

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Default Value	Suggestion
csi_driver.nodeLogging.fileDir	Directory of the node log in **file** output mode.	Yes	/var/log/huawei	Ensure that the directory has sufficient space for storing logs. It is recommended that the space be greater than or equal to 200 MB.
csi_driver.nodeLogging.fileSize	Size of a single node log file in **file** output mode.	Yes	20M	-
csi_driver.nodeLogging.maxBackups	Maximum number of node log file backups in **file** output mode.	Yes	9	-

CAUTION

If Huawei CSI has been deployed in your container environment, ensure that the value of **csi_driver.driverName** is the same as that configured during previous deployment. Otherwise, existing volumes or snapshots provisioned by Huawei CSI in the system cannot be managed by the newly deployed Huawei CSI.

backends Configuration Items

The **backends** configuration items include the information of the storage backends to be managed by the CSI plug-in and the protocol and port information used for connecting to the storage backends. The storage backends can provide storage resources for the Kubernetes container platform, for example, providing PVs for Kubernetes using SAN or NAS.

Huawei CSI can connect to multiple storage backends and allows you to configure advanced features such as ALUA and HyperMetro for the storage backends. This part describes the **backends** configuration items in detail.

Set the following parameters for **backends**:

Copyright © Huawei Technologies Co., Ltd.

Table 4-4 backends configuration items

Parameter	Description	Mandatory	Suggestion
backends.storage	Storage service type. ● If enterprise storage provides SAN, set this parameter to **oceanstor-san**. ● If enterprise storage provides NAS, set this parameter to **oceanstor-nas**. ● If distributed storage provides SAN, set this parameter to **fusionstorage-san**. ● If distributed storage provides NAS, set this parameter to **fusionstorage-nas**.	Yes	One backend can provide only one storage service. If a single Huawei storage system can provide both SAN and NAS storage services, you can configure multiple backends and use different storage service types for each backend.
backends.name	Storage backend name. The value can contain uppercase letters, lowercase letters, digits, and the following special characters: [-_] If multiple storage backends need to be configured, ensure that the storage backend name is unique.	Yes	-
backends.vstoreName	vStore name on the storage side. This parameter needs to be specified when the connected backend is OceanStor V3/V5 and resources need to be provisioned under a specified vStore.	Conditionally mandatory	This parameter needs to be specified only when the backend is OceanStor V3/V5 and vStores need to be supported.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Suggestion
backends.accountName	Account name on the storage side. This parameter is mandatory when OceanStor Pacific series NAS is connected and NAS resources need to be provisioned under a specified account.	Conditionally mandatory	This parameter needs to be specified only when the backend is OceanStor Pacific series NAS and accounts need to be supported.
backends.urls	Management URLs of storage device. The value format is a list. The value can be a domain name or an IP address + port number. Only IPv4 addresses are supported.	Yes	If the connected backend is OceanStor V6 or OceanStor Dorado V6 and resources need to be provisioned under a specified vStore, set this parameter to the URL of the logical management port of the vStore.
backends.pools	Storage pools of storage devices. The value format is a list.	Yes	-

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Suggestion
backends.parameters.protocol	Storage protocol. The value is a character string. • iscsi • fc • roce • fc-nvme • nfs • dpc • scsi	Yes	• If the value is set to **iscsi**, ensure that an iSCSI client has been installed on the connected compute node. • If the value is set to **nfs**, ensure that an NFS client tool has been installed on the connected compute node. • If the value is set to **fc-nvme** or **roce**, ensure that the nvme-cli tool of version 1.9 or later has been installed on the connected compute node. • If the value is set to **dpc**, ensure that DPC has been installed on the connected compute node and the node has been added as a DPC compute node on the storage device to be connected. • If the value is set to **scsi**, ensure that a distributed storage VBS client has been installed on the connected compute node.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Suggestion
backends.parameters.portals	Service access port. Nodes will use this port to read and write storage resources. The value is a character string. Multiple ports can be configured if the protocol is **iscsi** or **roce**. Only one port can be configured if the protocol is **nfs**. Service ports do not need to be configured if the protocol is **fc**, **fc-nvme**, or **dpc**. If the protocol is **scsi**, the port is in dictionary format where the key indicates the host name and the value indicates the IP address (only IPv4 addresses are supported).	Conditionally mandatory	If a vStore or account is used to connect to a backend, **portals** must be set to the logical port information of the vStore or account.
backends.parameters.ALUA	ALUA configuration of the storage backend. If the worker node uses the native multipathing software provided by the OS and ALUA is enabled, you need to configure this parameter.	Conditionally mandatory	If ALUA is enabled for the host multipathing software, ensure that the backend ALUA configuration is the same as that of the host ALUA configuration. For details about the ALUA configuration, see **8.1.1 Configuring ALUA Using Helm**.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Mandatory	Suggestion
backends.metrovStorePairID	HyperMetro vStore pair ID. This parameter is mandatory when a PV to be created on the storage side needs to support the NAS HyperMetro feature. In this case, you need to enter the ID of the HyperMetro vStore pair to which the PV to be created belongs.	Conditionally mandatory	You can query the HyperMetro vStore pair ID on DeviceManager. If the connected storage is OceanStor V3/V5, you also need to specify the **vstoreName** parameter.
backends.metroBackend	Backend name of the HyperMetro peer. The value is a character string. This parameter is mandatory when a PV to be created on the storage side needs to support the NAS HyperMetro feature. In this case, you need to enter the name of the other backend to form a HyperMetro pair with the current backend.	Conditionally mandatory	The names of the two backends in the pair must be entered. After the two backends form a HyperMetro relationship, they cannot form a HyperMetro relationship with other backends.
backends.supportedTopologies	Storage topology awareness configuration. The parameter format is JSON of the list type.	Conditionally mandatory	This parameter is mandatory if storage topology awareness is enabled. For details, see **8.2.1 Configuring Storage Topology Awareness Using Helm**.

Configuring Multiple Huawei Storage Backends

If multiple Huawei storage devices are used in a Kubernetes cluster or one Huawei storage device provides multiple storage service types, refer to the following configuration example.

```
# An array of storages with the access info
backends:
  - storage: "oceanstor-nas"
    name: "nfs-155"
    urls:
      - "https://192.168.129.155:8088"
```


Copyright © Huawei Technologies Co., Ltd.

```
  pools:
    - "StoragePool001"
  parameters:
    protocol: "nfs"
    portals:
      - "192.168.128.155"

- storage: "oceanstor-san"
  name: "iscsi-155"
  urls:
    - "https://192.168.129.155:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "iscsi"
    portals:
      - "192.168.128.156"
      - "192.168.128.157"

- storage: "fusionstorage-san"
  name: "iscsi-156"
  urls:
    - "https://192.168.129.156:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "iscsi"
    portals:
      - "192.168.128.160"
      - "192.168.128.161"

- storage: "fusionstorage-nas"
  name: "nfs-156"
  urls:
    - "https://192.168.129.156:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "nfs"
    portals:
      - "192.168.128.170"
```

Configuring Huawei Storage Backends for Multiple vStores or Accounts

If the connected Huawei storage is OceanStor V3/V5 series and vStores are required to connect to the storage for resource isolation, you need to configure the **vstoreName** parameter for each backend. In this example, the two storage backends are the same Huawei OceanStor Dorado V5 storage, but different vStores are used for connection. In this case, **portals** must be set to the logical port information owned by the vStores.

```
# An array of storages with the access info
backends:
  - storage: "oceanstor-nas"
    name: "nfs-vstore001"
    vstoreName: "vstore001"
    urls:
      - "https://192.168.129.155:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "nfs"
      portals:
        - "10.168.129.100"

  - storage: "oceanstor-nas"
    name: "nfs-vstore002"
```


Copyright © Huawei Technologies Co., Ltd.

```
  vstoreName: "vstore002"
  urls:
    - "https://192.168.129.155:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "nfs"
    portals:
      - "10.168.129.101"
```

If the connected Huawei storage is OceanStor V6 or OceanStor Dorado V6 series and vStores are required to connect to the storage for resource isolation, you need to configure the URL of the logical management port of a specified vStore for each backend. In this example, the two storage backends are the same Huawei OceanStor Dorado V6 storage, but different vStores are used for connection. In this case, **portals** must be set to the logical port information owned by the vStores.

```
# An array of storages with the access info
backends:
  - storage: "oceanstor-nas"
    name: "nfs-vstore001"
    urls:
      - "https://192.168.129.155:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "nfs"
      portals:
        - "10.168.129.100"

  - storage: "oceanstor-nas"
    name: "nfs-vstore002"
    urls:
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "nfs"
      portals:
        - "10.168.129.101"
```

If the connected Huawei storage is OceanStor Pacific series and accounts are required to connect to the storage for resource isolation, you need to configure the **accountName** parameter for each backend. In this example, the two storage backends are the same Huawei OceanStor Pacific storage, but different accounts are used for connection. In this case, **portals** must be set to the logical port information owned by the accounts.

```
# An array of storages with the access info
backends:
  - storage: "fusionstorage-nas"
    name: "nfs-account001"
    accountName: "***"
    urls:
      - "https://192.168.129.155:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "nfs"
      portals:
        - "10.168.129.100"

  - storage: "fusionstorage-nas"
    name: "nfs-account002"
   accountName: "***"
    urls:
```

 Copyright © Huawei Technologies Co., Ltd.

```
    - "https://192.168.129.156:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "nfs"
    portals:
      - "10.168.129.101"
```

Configuring a Storage Backend of the iSCSI Type

This example shows how to configure a backend of the iSCSI type for Huawei enterprise storage and distributed storage.

```
backends:
  - storage: "oceanstor-san"
    name: "dorado-iscsi-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "iscsi"
      portals:
        - "192.168.128.120"
        - "192.168.128.121"

  - storage: "fusionstorage-san"
    name: "pacific-iscsi-125"
    urls:
      - "https://192.168.129.125:28443"
      - "https://192.168.129.126:28443"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "iscsi"
      portals:
        - "192.168.128.122"
        - "192.168.128.123"
```

Configuring a Storage Backend of the FC Type

This example shows how to configure a backend of the FC type for Huawei enterprise storage.

```
backends:
  - storage: "oceanstor-san"
    name: "fc-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "fc"
```

Configuring a Storage Backend of the NVMe over RoCE Type

This example shows how to configure a backend of the NVMe over RoCE type for Huawei enterprise storage.

```
backends:
  - storage: "oceanstor-san"
    name: "roce-155"
    urls:
      - "https://192.168.129.155:8088"
```

 Copyright © Huawei Technologies Co., Ltd.

```
    - "https://192.168.129.156:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "roce"
    portals:
      - "192.168.128.120"
      - "192.168.128.121"
```

Configuring a Storage Backend of the NVMe over FC Type

This example shows how to configure a backend of the NVMe over FC type for Huawei enterprise storage.

```
backends:
  - storage: "oceanstor-san"
    # support upper&lower characters, numeric and [-_].
    name: "fc-nvme-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "fc-nvme"
```

Configuring a Storage Backend of the NFS Type

This example shows how to configure a backend of the NFS type for Huawei enterprise storage and distributed storage.

```
backends:
  - storage: "oceanstor-nas"
    name: "nfs-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "nfs"
      portals:
        - "192.168.128.155"

  - storage: "fusionstorage-nas"
    name: "nfs-126"
    urls:
      - "https://192.168.129.125:28443"
      - "https://192.168.129.126:28443"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "nfs"
      portals:
        - "192.168.128.123"
```

Configuring a Storage Backend of the SCSI Type

This example shows how to configure a backend of the SCSI type for Huawei distributed storage.

```
backends:
  - storage: "fusionstorage-san"
    name: "scsi-155"
    urls:
      - "https://192.168.129.155:28443"
```

 Copyright © Huawei Technologies Co., Ltd.

```
  pools:
    - "StoragePool001"
  parameters:
    protocol: "scsi"
    portals:
      - {"hostname01": "192.168.125.21"}
      - {"hostname02": "192.168.125.22"}
```

Configuring a Storage Backend of the DPC Type

This example shows how to configure a backend of the DPC type for Huawei distributed storage.

```
backends:
  - storage: "fusionstorage-nas"
    name: "dpc-155"
    urls:
      - "https://192.168.129.155:28443"
      - "https://192.168.129.156:28443"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "dpc"
```

Configuring Storage Backends of the HyperMetro Type

CSI allows you to provision HyperMetro volumes of the NFS type on the storage side when connecting with OceanStor V6 or OceanStor Dorado V6. This example shows how to configure backends of the HyperMetro type for Huawei OceanStor V6 or OceanStor Dorado V6.

```
backends:
  - storage: "oceanstor-nas"
    name: "nfs-hypermetro-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    metrovStorePairID: "f09838237b93c000"
    metroBackend: "nfs-hypermetro-157"
    parameters:
      protocol: "nfs"
      portals:
        - "192.168.129.155"

  - storage: "oceanstor-nas"
    name: "nfs-hypermetro-157"
    urls:
      - "https://192.168.129.157:8088"
      - "https://192.168.129.158:8088"
    pools:
      - "StoragePool001"
    metrovStorePairID: "f09838237b93c000"
    metroBackend: "nfs-hypermetro-155"
    parameters:
      protocol: "nfs"
      portals:
        - "192.168.129.157"
```


Copyright © Huawei Technologies Co., Ltd.

NOTICE

- Before configuring NAS HyperMetro, you need to configure the HyperMetro relationship between two storage devices, including the remote device, HyperMetro domain, and the like. The HyperMetro domain of the file system can only work in HyperMetro mode. For details about the configuration operation, see the product documentation of the corresponding storage model.
- The accounts for connecting to NAS HyperMetro backends must be the administrator accounts of the storage vStores.

Other Configuration Items

Other configuration items include some features of the CSI plug-in or the policies for obtaining images.

Table 4-5 Other configuration items

Parameter	Description	Mandatory	Default Value
sidecarImagePullPolicy	Pull policy of the sidecar image.	Yes	IfNotPresent
huaweiImagePullPolicy	Pull policy of the huawei-csi image.	Yes	IfNotPresent
huaweiCsiController.replicas	Quantity of huawei-csi-controller. The value can only be **1**.	Yes	1
snapshot.enable	Whether to enable the snapshot feature.	Yes	true
resizer.enable	Whether to enable the capacity expansion feature.	Yes	true

NOTICE

If **snapshot.enable** is enabled, the parameter is set to **true**. In this case, when the **helm install** command is executed, the system automatically reads the volume snapshot CRD in the **helm/crd** directory and installs the snapshot CRD resource.

4.1.2 Installing Huawei CSI

Prerequisites

- A Huawei CSI image has been created and uploaded to the image repository or imported to all nodes by following the instructions provided in **3.3 Uploading a Huawei CSI Image**.

Copyright © Huawei Technologies Co., Ltd.

- The component images on which Huawei CSI installation and running depend have been uploaded to the image repository or imported to all nodes. For details, see **3.4 Checking the Images on Which CSI Depends**.
- The volume snapshot component CRD on which the running of Huawei CSI depends has been installed. For details, see **3.5 Checking Volume Snapshot-Dependent Components**.
- If you want to use multipathing to connect to Huawei storage, ensure that multipathing software has been installed on all compute nodes. For details, see **3.6 Checking the Host Multipathing Configuration**.
- Helm 3 has been installed on the container management platform.
- The **values.yaml** file required for installing CSI has been prepared. For details, see **4.1.1 Preparing the values.yaml File**.
- All worker nodes of Kubernetes communicate properly with the service IP address of the storage device to be connected. In iSCSI scenarios, the **ping** command can be used to verify the connectivity.
- Software clients required by the corresponding protocol, such as iSCSI and NFS clients, have been installed on all worker nodes of Kubernetes.
- The accounts required for connecting to Huawei CSI have been created on the Huawei storage to be connected. For details, see **3.7 Checking the Accounts on 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 Copy the **helm** directory in the Kubernetes CSI component package to any directory on the master node. For details about the Helm tool path, see **Table 3-1**.

Step 3 Go to the **helm/esdk** working directory.

```
# cd helm/esdk
```

Step 4 Run the **helm install** *helm-huawei-csi* **./ -n** *huawei-csi* **--create-namespace** command to install Huawei CSI.

In the preceding command, *helm-huawei-csi* indicates the custom Helm chart name, **./** indicates that the Helm project in the current directory is used, and *huawei-csi* indicates the custom Helm chart namespace.

```
# helm install helm-huawei-csi ./ -n huawei-csi --create-namespace
NAME: helm-huawei-csi
LAST DEPLOYED: Wed Jun  8 11:50:28 2022
NAMESPACE: huawei-csi-helm-chart
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Step 5 Copy the **secretGenerate** tool in the Huawei CSI component package to any directory on the master node. For details about the tool path, see **Table 3-1**.

Step 6 Use an encryption tool to enter the user name and password of the storage device.

 Copyright © Huawei Technologies Co., Ltd.

NOTICE

- If you connect to an OceanStor Dorado V6 or OceanStor V6 storage backend using a vStore, use the vStore administrator account information as the user name and password of the storage device.
- Huawei CSI stores the user name and password of a storage device in the **secret** resource of Kubernetes. For details about how to improve the security of the **secret** resource, see the official Kubernetes document at **encrypt-data**.
- All Kubernetes resources in Huawei CSI are created in declarative object configuration mode. This operation sets the **kubectl.kubernetes.io/last-applied-configuration: '{...}'** annotation on each object. The annotation value contains the content of the configuration file used to create an object. Community description: **https://github.com/pulumi/pulumi-kubernetes/issues/1118**

1. Run the **chmod +x secretGenerate** command to grant the execute permission on the secretGenerate tool.
```
# chmod +x secretGenerate
```
2. Run the **./secretGenerate --namespace=***huawei-csi* **--logFileDir=***/var/log/huawei* command to run the secretGenerate tool. Change the value of **namespace** to the actual namespace. If this parameter value is not used, the **huawei-csi** namespace is used by default. Change the value of **logFileDir** to the actual log directory. If this parameter value is not used, the **/var/log/huawei** directory is used by default. Then enter the ID of the backend to be configured as prompted. If **Configured** is **false**, the backend is not configured. If **Configured** is **true**, the backend is configured.
```
# ./secretGenerate
Getting backend configuration information.............
--------------------------------------------------------------------------------
Number  Configured  BackendName        Urls
1       false       iscsi-25       [https://192.168.125.25:8088]
2       false       nfs-26          [https://192.168.125.26:8088]
3       false       dpc-27          [https://192.168.125.27:8088]
--------------------------------------------------------------------------------
Please enter the backend number to configure (Enter 'exit' to exit):3
```
3. Enter the user name and password as prompted to create a **secret** object.
```
*************************Current backend information**************************
Name:strage-backend-03
Urls:[https://192.168.125.27:8088]
Please enter this backend user name:admin
Please enter this backend password:
Verifying user name and password. Please wait.....................
The acount information of the backend strage-backend-03 has been configured successfully.
```
4. After the configuration is complete, enter **exit** to exit and save the configuration.
```
Please enter the backend number to configure (Enter 'exit' to exit):exit
Saving configuration. Please wait.............
The configuration is saved successfully.
```
5. Run the **kubectl get secret -n huawei-csi | grep huawei-csi-secret** command to check whether the **secret** object is successfully created.
```
# kubectl -n huawei-csi get secret huawei-csi-secret
NAME                TYPE     DATA   AGE
huawei-csi-secret   Opaque   1      8d
```

Step 7 After the huawei-csi service is deployed, run the **kubectl get pod -n huawei-csi** command to check whether the service is started.

 Copyright © Huawei Technologies Co., Ltd.

```
# kubectl get pod -n huawei-csi
NAME                                  READY   STATUS    RESTARTS   AGE
huawei-csi-controller-764bd64c97-kr2nm   7/7     Running   0          144m
huawei-csi-node-7m48s                 3/3     Running   0          144m
```

----End

4.2 Manually Compiling ConfigMap Files to Install Huawei CSI

This section describes how to manually compile all resource configuration files (such as **Config.yaml**) to install Huawei CSI without Helm. If you have installed CSI by referring to **4.1 Installing Huawei CSI Using Helm**, skip this section.

CAUTION

It is strongly recommended that you use Helm to install, deploy, and upgrade Huawei CSI. Manually compiling resource configuration files to install Huawei CSI will not be supported in later versions.

4.2.1 Creating ConfigMap Files Required for Running Huawei CSI

A ConfigMap is an API object used to save non-confidential data to key-value pairs. When Huawei CSI starts running, the information contained in the ConfigMap created in this section needs to be used as important running parameters.

4.2.1.1 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 service IP address of the storage device to be connected. (Huawei CSI uses the **ping** command to check.)
- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes. For details, see **3.6 Checking the Host Multipathing Configuration**.
- 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.

Copyright © Huawei Technologies Co., Ltd.

Step 2 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-6**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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 4-6 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-7**.

Table 4-7 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
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.
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.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in scenarios where iSCSI is used. This parameter is mandatory.	In scenarios where iSCSI 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. Take OceanStor Dorado 6.*x* series as an example. On DeviceManager, choose **Services** > **Network** > **Logical Ports** and obtain the IP address whose data protocol is iSCSI. (For other series, see the corresponding operation description.)

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.2.1.2 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

 Copyright © Huawei Technologies Co., Ltd.

referring to **10.4 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

- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes of Kubernetes. For details, see **3.6 Checking the Host Multipathing Configuration**.
- 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-8**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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"}
            }
        ]
    }
```


Copyright © Huawei Technologies Co., Ltd.

Table 4-8 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-9**.

Table 4-9 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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.
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.
parameters	Dictionary	Variable parameters in scenarios where FC is used. This parameter is mandatory.	In scenarios where FC is used, set the **protocol** parameter to a fixed value: **fc**.

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

 Copyright © Huawei Technologies Co., Ltd.

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

----End

4.2.1.3 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.
- 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-10**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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"]}
            }
        ]
    }
```


Copyright © Huawei Technologies Co., Ltd.

Table 4-10 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-11**.

Table 4-11 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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.
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.

 Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in scenarios where NFS 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. You can log in to DeviceManager to obtain the logical port IP address. Take OceanStor Dorado 6.*x* series as an example. On DeviceManager, choose **Services** > **Network** > **Logical Ports** and obtain the IP address whose data protocol is NFS. (For other series, see the corresponding operation description.)

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.2.1.4 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 service IP address of the storage device to be connected. (Huawei CSI uses the **ping** command to check.)
- The nvme-cli tool has been installed on all worker nodes of Kubernetes, and the tool version is 1.9 or later.

 Copyright © Huawei Technologies Co., Ltd.

- 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 of Kubernetes. For details, see **3.6 Checking the Host Multipathing 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-12**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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 4-12 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-13**.

Copyright © Huawei Technologies Co., Ltd.

Table 4-13 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
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.
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.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in scenarios where the NVMe over RoCE protocol is used. This parameter is mandatory.	In scenarios where the NVMe over 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. You can log in to DeviceManager to obtain the logical port IP address. Take OceanStor Dorado 6.*x* series as an example. On DeviceManager, choose **Services** > **Network** > **Logical Ports** and obtain the IP address whose data protocol is NVMe over RoCE. (For other series, see the corresponding operation description.)

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

4.2.1.5 Connecting to Enterprise Storage SAN over NVMe over FC

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

Restrictions

To connect to enterprise storage SAN over NVMe 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 **10.4 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

- The nvme-cli tool has been installed on all worker nodes of Kubernetes, and the tool version is 1.9 or later.

Copyright © Huawei Technologies Co., Ltd.

- If a multipathing network is used, ensure that multipathing software has been installed on all worker nodes of Kubernetes. For details, see **3.6 Checking the Host Multipathing Configuration**.
- 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-fc-nvme.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 4-14**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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-nvme"}
            }
        ]
    }
```

Table 4-14 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-15**.

Copyright © Huawei Technologies Co., Ltd.

Table 4-15 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
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.
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.
parameters	Dictionary	Variable parameters in scenarios where the NVMe over FC protocol is used. This parameter is mandatory.	In scenarios where the NVMe over FC protocol is used, set the **protocol** parameter to a fixed value: **fc-nvme**.

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

Copyright © Huawei Technologies Co., Ltd.

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

----End

4.2.1.6 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.
- 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-16**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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"}]}
            }
        ]
    }
```

 Copyright © Huawei Technologies Co., Ltd.

Table 4-16 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-17**.

Table 4-17 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	For FusionStorage, only one management URL can be configured.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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.
parameters	Dictionary	Variable parameters in scenarios where SCSI is used. This parameter is mandatory.	The **protocol** parameter is fixed to **scsi**. Set **portals** to a pair list of host names and VBS node IP addresses. 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 (only IPv4 addresses are supported currently). If there are multiple worker nodes, configure them in dictionary format and separate them with commas (,). In the preceding example, **hostname01** is the host name of a worker node in Kubernetes, and **192.168.125.21** is the management IP address of a VBS node after VBS is created for the worker node.

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

 Copyright © Huawei Technologies Co., Ltd.

4.2.1.7 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 service IP address of the storage device to be connected. (Huawei CSI uses the **ping** command to check.)
- 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 of Kubernetes. For details, see **3.6 Checking the Host Multipathing Configuration**.

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 8.0.0 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-18**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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"]}
            }
        ]
    }
```


Copyright © Huawei Technologies Co., Ltd.

Table 4-18 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-19**.

Table 4-19 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	For FusionStorage, only one management URL can be configured.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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.
parameters	Dictionary	Variable parameters in scenarios where iSCSI is used. This parameter is mandatory.	In scenarios where iSCSI 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. You can log in to DeviceManager to obtain the iSCSI service IP addresses. Take OceanStor Pacific series as an example. On DeviceManager, choose **Resources** > **Access** > **Service Network**. (For other series, see the corresponding operation description.)

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

 Copyright © Huawei Technologies Co., Ltd.

4.2.1.8 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.
- 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-20**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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"]}
            }
        ]
    }
```


Copyright © Huawei Technologies Co., Ltd.

Table 4-20 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-21**.

Table 4-21 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	For FusionStorage, only one management URL can be configured.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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.
parameters	Dictionary	Variable parameters in scenarios where NFS 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. You can log in to DeviceManager to obtain the logical port IP address. Take OceanStor Pacific series as an example. On DeviceManager, choose **Resources** > **Access** > **Service Network** and click the name of a zone. On the page that is displayed, click the **IP Address Pool** tab. The **IP Address/Mask** column indicates the logical port IP address. For other series, see the corresponding operation description.

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | grep huawei-csi-configmap** command to check whether the creation is successful. If the following information is displayed, the creation is successful.

 Copyright © Huawei Technologies Co., Ltd.

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

----End

4.2.1.9 Connecting to Distributed Storage NAS over DPC

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

Prerequisites

- All worker nodes of Kubernetes have been added as DPC compute nodes 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 Run the **vi** *huawei-csi-configmap.yaml* command to create a file named *huawei-csi-configmap.yaml*.

```
# vi huawei-csi-configmap.yaml
```

Step 3 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 **deploy/huawei-csi-configmap/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 4-22**.

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


Copyright © Huawei Technologies Co., Ltd.

Table 4-22 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 4-23**.

Table 4-23 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). **NOTE** If multiple storage backends need to be configured, ensure that the storage backend name is unique.
urls	List	Management URL of the storage device to be connected. This parameter is mandatory.	For FusionStorage, only one management URL can be configured.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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 pool name.
parameters	Dictionary	Variable parameters in scenarios where DPC is used. This parameter is mandatory.	In scenarios where DPC is used, set the **protocol** parameter to a fixed value: **dpc**.

Step 4 Run the **kubectl create -f** *huawei-csi-configmap.yaml* command to create *huawei-csi-configmap*.

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

Step 5 After the creation is complete, run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.2.1.10 Connecting to Huawei Storage Using vStores or Accounts

If you need to use vStores or accounts to connect to Huawei storage for resource isolation, you need to configure the vStore or account information for each backend. If the vStore or account information is not configured, the CSI system will use the default vStore or account of the storage system to create 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** *huawei-csi-configmap.yaml* command to modify the .yaml file.

- If the vStore uses OceanStor V3/V5 storage, add the **vstoreName** parameter to the backend configuration and set the parameter to the vStore name of the storage device.

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

 Copyright © Huawei Technologies Co., Ltd.

```
  }]
}
```

- If the vStore uses OceanStor 6.1 or OceanStor Dorado 6.*x* storage, configure the URL of the logical management port of a specified vStore for each backend. Note that **urls** indicates the logical management ports of the vStore, and **pools** and **portals** must be available storage pools and logical data ports of the current vStore respectively.
- If the vStore uses OceanStor Pacific series storage, add the **accountName** parameter to the backend configuration and set the parameter to the account name to be specified. In this case, the backend **portals** must be set to the logical port owned by the account.

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

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 Run the **kubectl get configmap -n huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap                1      5s
```

----End

4.2.1.11 Provisioning HyperMetro Volumes at Backends Using CSI

CSI allows you to provision HyperMetro volumes of the NFS type on the storage side when connecting with Huawei OceanStor Dorado V6 or OceanStor V6. Perform this operation when you want to configure HyperMetro backends.

NOTICE

- Before configuring NAS HyperMetro, you need to configure the HyperMetro relationship between two storage devices, including the remote device, HyperMetro domain, and the like. The HyperMetro domain of the file system can only work in HyperMetro mode. For details about the configuration operation, see the product documentation of the corresponding storage model.
- The accounts for connecting to NAS HyperMetro backends must be the administrator accounts of the storage vStores.

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. For details about the configuration items for each

 Copyright © Huawei Technologies Co., Ltd.

backend, see **4.2.1.3 Connecting to Enterprise Storage NAS over NFS**. You need to add some additional configuration parameters in the HyperMetro scenario. For details, see **Table 4-24**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-nas",
                "name": "hyperMetro1",
                "urls": ["https://192.168.125.20:8088", "https://192.168.125.21:8088"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "nfs", "portals": ["192.168.125.22"]},
                "metrovStorePairID": "f09838237b93c000",
                "metroBackend": "hyperMetro2"
            },
            {
                "storage": "oceanstor-nas",
                "name": "hyperMetro2",
                "urls": ["https://192.168.125.24:8088", "https://192.168.125.25:8088"],
                "pools": ["storagepool01", "storagepool02"],
                "parameters": {"protocol": "nfs", "portals": ["192.168.125.26"]},
                "metrovStorePairID": "f09838237b93c000",
                "metroBackend": "hyperMetro1"
            }
        ]
    }
```

Table 4-24 HyperMetro configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
vstoreName	String	vStore name. This parameter is conditionally mandatory (mandatory for OceanStor V3/V5).	Only vStore users support NAS HyperMetro.
metrovStorePairID	String	ID of the HyperMetro vStore pair to which a vStore belongs. This parameter is mandatory.	For example, the parameter of OceanStor Dorado 6.*x* or OceanStor 6.1 is displayed as **ID** on DeviceManager.
metroBackend	String	Name of a peer end in HyperMetro. The two backends form a HyperMetro relationship. This parameter is mandatory.	The peer end of **hyperMetro1** is **hyperMetro2**, and the peer end of **hyperMetro2** is **hyperMetro1**.

Copyright © Huawei Technologies Co., Ltd.

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 huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

4.2.1.12 Connecting to Multiple Backends Using CSI

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 **4.2.1 Creating ConfigMap Files Required for Running Huawei CSI**.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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 huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

----End

Copyright © Huawei Technologies Co., Ltd.

4.2.2 Starting huawei-csi Services

This section describes how to start huawei-csi services.

Prerequisites

- A Huawei CSI image has been created and uploaded to the image repository or imported to all nodes by following the instructions provided in **3.3 Uploading a Huawei CSI Image**.
- The component images on which Huawei CSI installation and running depend have been uploaded to the image repository or imported to all nodes. For details, see **3.4 Checking the Images on Which CSI Depends**.
- The volume snapshot component CRD on which the running of Huawei CSI depends has been installed. For details, see **3.5 Checking Volume Snapshot-Dependent Components**.
- If you want to use multipathing to connect to Huawei storage, ensure that multipathing software has been installed on all compute nodes. For details, see **3.6 Checking the Host Multipathing Configuration**.
- All worker nodes of Kubernetes communicate properly with the service IP address of the storage device to be connected. In iSCSI scenarios, the **ping** command can be used to verify the connectivity.
- Software clients required by the corresponding protocol, such as iSCSI and NFS clients, have been installed on all worker nodes of Kubernetes.
- The accounts required for connecting to Huawei CSI have been created on the Huawei storage to be connected. For details, see **3.7 Checking the Accounts on 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 Copy the **secretGenerate** tool in the Kubernetes CSI component package to any directory on the master node. For details about the tool path, see **Table 3-1**.

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

Copyright © Huawei Technologies Co., Ltd.

NOTICE

- If you connect to an OceanStor Dorado V6 or OceanStor V6 storage backend using a vStore, use the vStore administrator account information as the user name and password of the storage device.
- Huawei CSI stores the user name and password of a storage device in the **secret** resource of Kubernetes. For details about how to improve the security of the **secret** resource, see the official Kubernetes document at **encrypt-data**.
- All Kubernetes resources in Huawei CSI are created in declarative object configuration mode. This operation sets the **kubectl.kubernetes.io/last-applied-configuration: '{...}'** annotation on each object. The annotation value contains the content of the configuration file used to create an object. Community description: **https://github.com/pulumi/pulumi-kubernetes/issues/1118**

1. Run the **chmod +x secretGenerate** command to grant the execute permission on the secretGenerate tool.
```
# chmod +x secretGenerate
```
2. Run the **./secretGenerate --namespace=***huawei-csi* **--logFileDir=***/var/log/huawei* command to run the secretGenerate tool. Change the value of **namespace** to the actual namespace. If this parameter value is not used, the **huawei-csi** namespace is used by default. Change the value of **logFileDir** to the actual log directory. If this parameter value is not used, the **/var/log/huawei** directory is used by default. Then enter the ID of the backend to be configured as prompted. If **Configured** is **false**, the backend is not configured. If **Configured** is **true**, the backend is configured.
```
# ./secretGenerate huawei-csi
Getting backend configuration information.....
--------------------------------------------------------------------------------
Number  Configured  BackendName        Urls
1       false       strage-backend     [https://192.168.125.25:8088]
2       false       strage-backend-02  [https://192.168.125.26:8088]
3       false       strage-backend-03  [https://192.168.125.27:8088]
4       false       strage-backend-04  [https://192.168.125.28:8088]
5       false       strage-backend-05  [https://192.168.125.29:28443]
6       false       strage-backend-06  [https://192.168.125.30:28443]
--------------------------------------------------------------------------------
Please enter the backend number to configure (Enter 'exit' to exit):3
```
3. Enter the user name and password as prompted to create a **secret** object.
```
**************************Current backend information**************************
Name:strage-backend-03
Urls:[https://192.168.125.27:8088]
Please enter this backend user name:admin
Please enter this backend password:
Verifying user name and password. Please wait.....
The acount information of the backend strage-backend-03 has been configured successfully.
```
4. After the configuration is complete, enter **exit** to exit and save the configuration.
```
Please enter the backend number to configure (Enter 'exit' to exit):exit
Saving configuration. Please wait.....
The configuration is saved successfully.
```
5. Run the **kubectl get secret -n huawei-csi | grep huawei-csi-secret** command to check whether the **secret** object is successfully created.
```
# kubectl -n huawei-csi get secret huawei-csi-secret
NAME              TYPE     DATA   AGE
huawei-csi-secret   Opaque   1      8d
```

 Copyright © Huawei Technologies Co., Ltd.

Step 4 Run the following command to create the RBAC permission.

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

Step 5 Start the controller service.

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

   NOTE

   - (Mandatory) In the **image** configuration item under **huawei-csi-driver** in the sample .yaml file, change **huawei-csi:****.*.** to *<Name>***:***<Version>* of the Huawei CSI image uploaded in **3.3 Uploading a Huawei CSI Image**.

     ```
     containers:
       ...
       - name: huawei-csi-driver
         image: repo.huawei.com/huawei-csi:3.1.0
     ```

   - (Optional) The **namespace** configuration item under **metadata** in the sample .yaml file indicates the namespace where the huawei-csi-controller service is installed. If you need to change the value, ensure that the namespaces in the **config.yaml**, **rbac.yaml**, and **node.yaml** files are the same. The change method is as follows.

     ```
     metadata:
       ...
       name: huawei-csi-controller
       namespace: huawei-csi
     ```

2. Run the following command to start the controller service.

   ```
   # kubectl apply -f huawei-csi-controller-snapshot-v1.yaml
   ```

Step 6 Start the node service.

1. 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.

Copyright © Huawei Technologies Co., Ltd.

NOTE

- (Mandatory) In the **image** configuration item under **huawei-csi-driver** in the sample .yaml file, change **huawei-csi:****.*.** to *<Name>*:*<Version>* of the Huawei CSI image uploaded in **3.3 Uploading a Huawei CSI Image**.

```
containers:
  ...
  - name: huawei-csi-driver
    image: huawei-csi:3.1.0
```

- (Optional) The **namespace** configuration item under **metadata** in the sample .yaml file indicates the namespace where the huawei-csi-node service is installed. If you need to change the value, ensure that the namespaces in the **config.yaml**, **rbac.yaml**, and **controller.yaml** files are the same. The change method is as follows.

```
metadata:
  ...
  name: huawei-csi-controller
  namespace: huawei-csi
```

- (Optional) 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"
```

- (Optional) In the **args** section of **huawei-csi-driver** in the .yaml file, **--connector-threads** indicates the number of concurrent operations on the drive letter on the host. The value is an integer ranging from 1 to 10, and the default value is **4**. To change the value, refer to the following.

```
args:
  - "--endpoint=/csi/csi.sock"
  - "--containerized"
  - "--driver-name=csi.huawei.com"
  - "--volume-use-multipath=true"
  - "--connector-threads=5"
```

- (Optional) In the **args** section of **huawei-csi-driver** in the .yaml file, **--scan-volume-timeout** indicates the timeout of waiting for multipathing aggregation when DM-Multipath is used on the host. The value is an integer ranging from 1 to 600, and the default value is **3**. To change the value, refer to the following.

```
args:
  - "--endpoint=/csi/csi.sock"
  - "--containerized"
  - "--driver-name=csi.huawei.com"
  - "--volume-use-multipath=true"
  - "--connector-threads=4"
  - "--scan-volume-timeout=3"
```

- (Optional) In the **args** section of **huawei-csi-driver** in the .yaml file, for enterprise storage, if **--volume-use-multipath** is set to **true**, you can configure the multipathing type according to the networking mode. For details, see **Table 4-25**.

```
args:
  - "--endpoint=/csi/csi.sock"
  - "--containerized"
  - "--driver-name=csi.huawei.com"
  - "--connector-threads=4"
  - "--volume-use-multipath=true"
  - "--scsi-multipath-type=DM-multipath"
  - "--nvme-multipath-type=HW-UltraPath-NVMe"
```


Copyright © Huawei Technologies Co., Ltd.

Table 4-25 Parameters for configuring enterprise storage multipathing

Storage Protocol	Parameter	Description	Remarks
iSCSI/FC	--scsi-multipath-type	The value can be: – DM-multipath – HW-UltraPath – HW-UltraPath-NVMe The default value is **DM-multipath**.	– **DM-multipath**: native multipathing software of the OS – **HW-UltraPath**: Huawei UltraPath multipathing software – **HW-UltraPath-NVMe**: Huawei UltraPath-NVMe multipathing software
NVMe over RoCE/NVMe over FC	--nvme-multipath-type	The default value is **HW-UltraPath-NVMe** and only **HW-UltraPath-NVMe** can be configured.	

2. Run the following command to start the node service.
```
# kubectl apply -f huawei-csi-node.yaml
```

Step 7 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
huawei-csi   huawei-csi-controller-695b84b4d8-tg64l   7/7   Running   0   14s
huawei-csi   huawei-csi-node-g6f7z                    3/3   Running   0   14s
```

----End

 Copyright © Huawei Technologies Co., Ltd.

5 Uninstalling Huawei CSI

This chapter describes how to uninstall Huawei CSI. The uninstallation method varies according to the installation mode.

- If Huawei CSI is installed using Helm, see **5.1 Uninstalling huawei-csi Using Helm**.
- If Huawei CSI is installed by manually compiling resource configuration files, see **5.2 Manually Uninstalling huawei-csi**.

CAUTION

If you do not uninstall Huawei CSI for the purpose of an upgrade, ensure that all resources (such as PV, PVC, and snapshot resources) provisioned by Huawei CSI have been cleared on your container platform before uninstalling Huawei CSI. Otherwise, once you uninstall Huawei CSI, these resources cannot be automatically scheduled, managed, or cleared.

5.1 Uninstalling huawei-csi Using Helm
5.2 Manually Uninstalling huawei-csi
5.3 Uninstalling the Snapshot-Dependent Component Service

5.1 Uninstalling huawei-csi Using Helm

If Huawei CSI is installed using Helm, uninstall Huawei CSI by referring to this section.

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 **helm uninstall** *helm-huawei-csi* **-n** *huawei-csi* command to uninstall Huawei CSI. In the command, *helm-huawei-csi* indicates the custom Helm chart name and *huawei-csi* indicates the namespace where the Helm chart resides. This command will delete the huawei-csi-controller, huawei-csi-node, huawei-csi-configmap, and RBAC resources of Huawei CSI.

 Copyright © Huawei Technologies Co., Ltd.

```
# helm uninstall helm-huawei-csi -n huawei-csi
release "helm-huawei-csi" uninstalled
```

After the deletion command is executed, you need to check whether the uninstallation is successful.

```
# helm list -n huawei-csi
NAME    NAMESPACE       REVISION        UPDATED STATUS  CHART   APP VERSION
```

In the preceding command, *huawei-csi* indicates the namespace where the chart is located.

If the command output is empty, the service is successfully deleted.

Step 3 Delete the **secret** object.

```
# kubectl delete secret huawei-csi-secret -n huawei-csi
```

In the preceding command, *huawei-csi-secret* indicates the name of the **secret** object, and *huawei-csi* indicates the namespace where the **secret** object is located.

After the deletion command is executed, you need to check whether the deletion is successful.

```
# kubectl get secret huawei-csi-secret -n huawei-csi
Error from server (NotFound): secrets "huawei-csi-secret" not found
```

If **NotFound** is displayed in the command output, the *huawei-csi-secret* object has been successfully deleted.

Step 4 (Optional) Uninstall the snapshot-dependent component service. For details, see **5.3 Uninstalling the Snapshot-Dependent Component Service**.

Step 5 (Optional) Delete the Huawei CSI image.

----End

5.2 Manually Uninstalling huawei-csi

If Huawei CSI is installed by manually compiling resource configuration files, uninstall Huawei CSI by referring to this section.

5.2.1 Uninstalling 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 **kubectl delete daemonset huawei-csi-node -n** *huawei-csi* command to uninstall the huawei-csi-node service. Replace *huawei-csi* with the namespace where Huawei CSI is located.

```
# kubectl delete daemonset huawei-csi-node -n huawei-csi
```

Step 3 Run the following command to check whether the service is successfully uninstalled. If **NotFound** is displayed, the service is successfully uninstalled.

```
# kubectl get daemonset huawei-csi-node -n huawei-csi
```

----End

Copyright © Huawei Technologies Co., Ltd.

5.2.2 Uninstalling 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 **kubectl delete deployment huawei-csi-controller -n** *huawei-csi* command to uninstall the huawei-csi-controller service. Replace *huawei-csi* with the namespace where Huawei CSI is located.

```
# kubectl delete deployment huawei-csi-controller -n huawei-csi
```

Step 3 Run the following command to check whether the service is successfully uninstalled. If **NotFound** is displayed, the service is successfully uninstalled.

```
# kubectl get deployment huawei-csi-controller -n huawei-csi
```

----End

5.2.3 Deleting the huawei-csi-configmap Object

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 delete configmap** *huawei-csi-configmap* **-n** *huawei-csi* command to delete the **configmap** object. *huawei-csi-configmap* is the name of the **configmap** object, and *huawei-csi* is the namespace where the **configmap** object is located.

```
# kubectl delete configmap huawei-csi-configmap -n huawei-csi
```

Step 3 Run the following command to check whether the object is successfully deleted. If **NotFound** is displayed, the object is successfully deleted.

```
# kubectl get configmap huawei-csi-configmap -n huawei-csi
```

----End

5.2.4 Deleting the huawei-csi-secret Object

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 delete secret** *huawei-csi-secret* **-n** *huawei-csi* command to delete the **secret** object. *huawei-csi-secret* is the name of the **secret** object, and *huawei-csi* is the namespace where the **secret** object is located.

```
# kubectl delete secret huawei-csi-secret -n huawei-csi
```

Step 3 Run the following command to check whether the **secret** object is successfully deleted. If **NotFound** is displayed in the command output, the **huawei-csi-secret** object is successfully deleted.

```
# kubectl get secret huawei-csi-secret -n huawei-csi
Error from server (NotFound): secrets "huawei-csi-secret" not found
```

----End

Copyright © Huawei Technologies Co., Ltd.

5.2.5 Deleting the RBAC Permission

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 Delete the RBAC permission.

- If the huawei-csi version is later than 2.2.15, run the following command to delete the permission. **-n** indicates the namespace. Change it based on site requirements.

```
# kubectl -n huawei-csi -l provisioner=csi.huawei.com delete
ServiceAccount,role,rolebinding,ClusterRole,ClusterRoleBinding
```

- If the huawei-csi version is 2.2.15 or earlier, perform the following operations to delete the permission.

  a. Run the following command to create a file named **delete-huawei-csi-rbac.sh**. **-n** indicates the namespace. Change it based on site requirements. For CSI earlier than V3.0, the default namespace is **kube-system**.

```
# cat <<EOF > delete-huawei-csi-rbac.sh
kubectl delete ServiceAccount huawei-csi-controller -n huawei-csi
kubectl delete ServiceAccount huawei-csi-node -n huawei-csi
kubectl delete ClusterRole huawei-csi-attacher-runner -n huawei-csi
kubectl delete ClusterRole huawei-csi-driver-registrar-runner -n huawei-csi
kubectl delete ClusterRole huawei-csi-provisioner-runner -n huawei-csi
kubectl delete ClusterRole huawei-csi-resizer-runner -n huawei-csi
kubectl delete ClusterRole huawei-csi-snapshotter-runner -n huawei-csi
kubectl delete ClusterRole snapshot-controller-runner -n huawei-csi
kubectl delete ClusterRoleBinding huawei-csi-attacher-role -n huawei-csi
kubectl delete ClusterRoleBinding huawei-csi-driver-registrar-role -n huawei-csi
kubectl delete ClusterRoleBinding huawei-csi-provisioner-role -n huawei-csi
kubectl delete ClusterRoleBinding huawei-csi-resizer-role -n huawei-csi
kubectl delete ClusterRoleBinding huawei-csi-snapshotter-role -n huawei-csi
kubectl delete ClusterRoleBinding snapshot-controller-role -n huawei-csi
kubectl delete Role huawei-csi-resizer-cfg -n huawei-csi
kubectl delete Role huawei-csi-snapshotter-leaderelection -n huawei-csi
kubectl delete Role snapshot-controller-leaderelection -n huawei-csi
kubectl delete RoleBinding huawei-csi-resizer-role-cfg -n huawei-csi
kubectl delete RoleBinding huawei-csi-snapshotter-leaderelection -n huawei-csi
kubectl delete RoleBinding snapshot-controller-leaderelection -n huawei-csi

EOF
```

  b. Run the following command to delete the RBAC permission. If the **NotFound** error is reported, ignore it.

```
# sh delete-huawei-csi-rbac.sh
```

Step 3 Check whether the RBAC permission has been deleted.

- If the huawei-csi version is later than 2.2.15, run the following command. **-n** indicates the namespace. Change it based on site requirements. If **No resources found** is displayed, the permission is successfully deleted.

```
# kubectl -n huawei-csi -l provisioner=csi.huawei.com get
ServiceAccount,role,rolebinding,ClusterRole,ClusterRoleBinding
```

- If the huawei-csi version is 2.2.15 or earlier, perform the following operations to check whether the RBAC permission is successfully deleted.

  a. Run the following command to create a file named **check-huawei-csi-rbac.sh**. **-n** indicates the namespace. Change it based on site requirements.

 Copyright © Huawei Technologies Co., Ltd.

```
# cat <<EOF > check-huawei-csi-rbac.sh
kubectl get ServiceAccount -n huawei-csi | grep huawei-csi
kubectl get ClusterRole -n huawei-csi | grep huawei-csi
kubectl get ClusterRoleBinding -n huawei-csi | grep huawei-csi
kubectl get Role -n huawei-csi | grep huawei-csi
kubectl get RoleBinding -n huawei-csi | grep huawei-csi
kubectl get ClusterRole snapshot-controller-runner -n huawei-csi --ignore-not-found=true
kubectl get ClusterRoleBinding snapshot-controller-role -n huawei-csi --ignore-not-found=true
kubectl get Role snapshot-controller-leaderelection -n huawei-csi --ignore-not-found=true
kubectl get RoleBinding snapshot-controller-leaderelection -n huawei-csi --ignore-not-found=true

EOF
```

b. Run the following command. If no command output is displayed, the RBAC permission has been successfully deleted.

```
# sh check-huawei-csi-rbac.sh
```

----End

5.2.6 Deleting the Image of the Earlier Version

To delete the **huawei-csi** image from the cluster, you need to perform the deletion operation on all worker nodes.

To delete the image from a single node, perform the following steps.

Prerequisites

The container service that depends on the image has been stopped. Otherwise, the image cannot be deleted.

Procedure

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

Step 2 Run the following command to view all existing versions.

- If docker is used, run the **docker image ls | grep huawei-csi** command.

```
# docker image ls | grep huawei-csi
REPOSITORY    TAG             IMAGE ID        CREATED          SIZE
huawei-csi    2.2.15          b30b3a8b5959    2 weeks ago      79.7MB
huawei-csi    3.1.0           14b854dba227    2 weeks ago      79.6MB
```

- If containerd is used, run the **crictl image ls | grep huawei-csi** command.

```
# crictl image ls | grep huawei-csi
REPOSITORY                    TAG             IMAGE ID        CREATED          SIZE
docker.io/library/huawei-csi  2.2.15          b30b3a8b5959    2 weeks ago      79.7MB
docker.io/library/huawei-csi  3.1.0           14b854dba227    2 weeks ago      79.6MB
```

Step 3 Run the following command to delete the image of the earlier version:

- If docker is used, run the **docker rmi** *<REPOSITORY>*:*<TAG>* command.

```
# docker rmi huawei-csi:2.2.15
```

- If containerd is used, run the **crictl rmi** *<REPOSITORY>*:*<TAG>* command.

```
# crictl rmi huawei-csi:2.2.15
```

Step 4 Run the following command again to check whether the image is successfully deleted. If the target version is not displayed, the image of the version is successfully deleted.

- If docker is used, run the **docker image ls | grep huawei-csi** command.

 Copyright © Huawei Technologies Co., Ltd.

```
# docker image ls | grep huawei-csi
huawei-csi                      3.1.0       14b854dba227   10 minutes ago   80MB
```

- If containerd is used, run the **crictl image ls | grep huawei-csi** command.

```
# crictl image ls | grep huawei-csi
docker.io/library/huawei-csi        3.1.0               14b854dba2273       93.1MB
```

----End

5.3 Uninstalling the Snapshot-Dependent Component Service

CAUTION

- Do not uninstall the snapshot-dependent component service when snapshots exist. Otherwise, Kubernetes will automatically delete all user snapshots and they cannot be restored. Exercise caution when performing this operation. For details, see **Delete a CustomResourceDefinition**.
- Do not uninstall the snapshot-dependent component service during the CSI upgrade.

Scenario Description

- Currently, Huawei CSI uses the snapshot feature.
- Currently, only Huawei CSI is available in the Kubernetes cluster, and Huawei CSI is no longer used.
- Before the uninstallation, ensure that no VolumeSnapshot resource managed by Huawei CSI exists in the Kubernetes cluster.

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 uninstall the snapshot-dependent component service.

```
# kubectl delete crd volumesnapshotclasses.snapshot.storage.k8s.io
volumesnapshotcontents.snapshot.storage.k8s.io volumesnapshots.snapshot.storage.k8s.io
```

Step 3 Run the following command to check whether the service is successfully uninstalled.

If the command output is empty, the uninstallation is successful.

```
# kubectl get crd | grep snapshot.storage.k8s.io
```

----End

Copyright © Huawei Technologies Co., Ltd.

6 Upgrade/Rollback Operations

This chapter describes how to upgrade or roll back Huawei CSI. The upgrade or rollback method varies according to the installation mode.

- If Huawei CSI is installed using Helm, see **6.1 Upgrading or Rolling Back Huawei CSI Using Helm**.
- If Huawei CSI is installed by manually compiling resource configuration files, see **6.2 Manually Compiling ConfigMap Files to Upgrade or Roll Back Huawei CSI**.

During the upgrade or rollback, the existing resources such as PVCs, snapshots, and Pods will run properly and will not affect your service access.

CAUTION

- During the upgrade or rollback, you cannot use Huawei CSI to create new resources or mount or unmount an existing PVC.
- During the upgrade or rollback, do not uninstall the snapshot-dependent component service.

6.1 Upgrading or Rolling Back Huawei CSI Using Helm

6.2 Manually Compiling ConfigMap Files to Upgrade or Roll Back Huawei CSI

6.1 Upgrading or Rolling Back Huawei CSI Using Helm

If Huawei CSI is installed using Helm, perform the upgrade or rollback by referring to this section.

6.1.1 Upgrading Huawei CSI

Prerequisites

When upgrading CSI, ensure that the parameter configurations in the **backends** field in the **values.yaml** file are the same as those in the **huawei-csi-configmap.yaml** file configured during Huawei CSI installation. Otherwise, CSI

 Copyright © Huawei Technologies Co., Ltd.

upon the upgrade cannot manage the previously provisioned resources such as PVCs and Pods.

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the file, and change the parameter values based on the upgrade requirements. After the change is complete, press **Esc** and enter **:wq!** to save the change. For details, see **4.1.1 Preparing the values.yaml File**.

```
# vi values.yaml
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* ./ **-n** *huawei-csi* **-f** *values.yaml* command to upgrade the Helm chart. The upgrade command will update the **Deployment**, **daemonset**, and **RBAC** resources, but will not update the **secret** resource. If the backend information changes, you must manually update **secret**. For details, see **9.1 Updating the User Name or Password of a Storage Device Configured on CSI**.

In the preceding command, *helm-huawei-csi* indicates the custom chart name and *huawei-csi* indicates the custom namespace.

```
# helm upgrade helm-huawei-csi ./  -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----End

6.1.2 Rolling Back CSI

The Helm rollback command rolls back the current version to the specified version. If no version number is specified, it will be rolled back to the previous version.

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 Check the version of Huawei CSI installed using Helm. You can run the **helm list -A** command to query all Helm versions in all namespaces, and the **helm list -n** *huawei-csi* command to query the Helm version in the specified namespace.

Copyright © Huawei Technologies Co., Ltd.

```
# helm list -A
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART
APP VERSION
helm-test       huawei-csi      1               2022-06-08 08:48:30.038729177 +0000 UTC deployed
esdk-1.0.0      3.1.0
# helm list -n huawei-csi
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART
APP VERSION
helm-test       huawei-csi      1               2022-06-08 08:48:30.038729177 +0000 UTC deployed
esdk-1.0.0      3.1.0
```

Step 3 Run the **helm rollback** *helm-huawei-csi revision-number* **-n** *huawei-csi* command to roll back Huawei CSI to the specified version. In the following example, the version is rolled back to **1**.

In the preceding command, *helm-huawei-csi* indicates the name of the chart to be rolled back, *revision-number* indicates the target version of the rollback, and *huawei-csi* indicates the namespace where the chart is located.

```
# helm rollback helm-huawei-csi 1 -n huawei-csi
Rollback was a success! Happy Helming!
```

----End

6.2 Manually Compiling ConfigMap Files to Upgrade or Roll Back Huawei CSI

If Huawei CSI is installed by manually compiling ConfigMap files, perform the upgrade or rollback by referring to this section. The upgrade or rollback includes the following operations:

- Uninstalling original CSI
- Installing new CSI

6.2.1 Uninstalling Original CSI

Before upgrading or rolling back Huawei CSI, uninstall the original Huawei CSI.

Before uninstalling CSI, run the **kubectl get configmap huawei-csi-configmap -n huawei-csi -o yaml > huawei-csi-configmap.yaml.bak** command to back up the content of the **huawei-csi-configmap** file. (During the CSI upgrade, the **backends** parameter in **huawei-csi-configmap.yaml** must be the same as the existing **configmap** configuration.)

CAUTION

Before the upgrade or rollback, back up all CSI configurations to prevent configuration loss caused by an upgrade failure.

After the backup is complete, uninstall the original CSI by referring to **5.2 Manually Uninstalling huawei-csi**.

6.2.2 Installing New CSI

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

Copyright © Huawei Technologies Co., Ltd.

Prerequisites

You have performed the operations described in **6.2.1 Uninstalling Original CSI** and backed up **huawei-csi-configmap.yaml** in the original CSI.

Procedure

Step 1 Prepare for the installation. For details, see **3 Installation Preparations**.

Step 2 Install the new Huawei CSI.

- If the version of Huawei CSI to be installed is 3.0.0 or later, you are strongly advised to install Huawei CSI using Helm. Install the CSI plug-in of the new version by referring to **4.1 Installing Huawei CSI Using Helm**.
- If the version of Huawei CSI to be installed is earlier than 3.0.0, install the CSI plug-in of the new version by referring to **4.2 Manually Compiling ConfigMap Files to Install Huawei CSI**.

CAUTION

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 that has been backed up.
- For details about **urls** and **parameters**, see the **huawei-csi-configmap.yaml.bak** file that has been backed up. Set these parameters as required.

----End

Copyright © Huawei Technologies Co., Ltd.

7 Using Huawei CSI

This chapter describes how to use Huawei CSI to manage the lifecycle of PVs and snapshots.

7.1 Managing a PV/PVC

7.2 Creating a VolumeSnapshot

7.1 Managing a PV/PVC

Based on service requirements, files in containers need to be persistently stored on disks. When the containers are re-built or re-allocated to new nodes, the persistent data can still be used.

To persistently store data on storage devices, you need to use the **PersistentVolume (PV)** and **PersistentVolumeClaim (PVC)** when provisioning containers.

- PV: a piece of storage in the Kubernetes cluster that has been provisioned by an administrator or dynamically provisioned using a **StorageClass**.
- PVC: a request for storage by a user. A PVC consumes PV resources. A PVC can request specific size and access modes. For example, a PV can be mounted in ReadWriteOnce, ReadOnlyMany, or ReadWriteMany mode. For details, see **Access Modes**.

This section describes how to use Huawei CSI to create, expand the capacity of, and clone a PV/PVC, as well as create a PVC using a snapshot.

7.1.1 Creating a PVC

Huawei CSI allows storage resources (LUNs or file systems) to be created on Huawei storage and provided for containers based on user settings. For details about the supported features, see **Table 2-4** or **Table 2-6**.

A PVC can be created in dynamic volume provisioning or static volume provisioning mode.

- Dynamic volume provisioning does not require a PV to be created in advance. Huawei CSI automatically creates resources required by a PV on storage

 Copyright © Huawei Technologies Co., Ltd.

devices based on a StorageClass. In addition, you can create a PV when creating a PVC.

- Static volume provisioning requires the administrator to create required resources on a storage device in advance and use existing resources by creating a PV. In addition, you can specify the associated PV when creating a PVC.

7.1.1.1 Dynamic Volume Provisioning

Dynamic volume provisioning allows storage volumes to be created on demand. Dynamic volume provisioning depends on the StorageClass objects. The cluster administrator can define multiple StorageClass objects as required and specify a StorageClass that meets service requirements when declaring a PV or PVC. When applying for resources from Huawei storage devices, Huawei CSI creates storage resources that meet service requirements based on the preset StorageClass.

To implement dynamic volume provisioning, perform the following steps:

- Configuring a StorageClass
- Configuring a PVC

7.1.1.1.1 Configuring a StorageClass

A **StorageClass** provides administrators with methods to describe a storage "class". Different types may map to a different group of capability definitions. Kubernetes cluster users can dynamically provision volumes based on a StorageClass.

A StorageClass supports the following parameters.

If SAN storage is used, refer to example file **/examples/sc-lun.yaml**. If NAS storage is used, refer to example file **/examples/sc-fs.yaml**.

Table 7-1 StorageClass configuration parameters

Parameter	Description	Remarks
metadata.name	User-defined name of a StorageClass object. This parameter is mandatory.	Take Kubernetes v1.22.1 as an example. The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit.
provisioner	Name of the provisioner. This parameter is mandatory.	Set this parameter to the driver name set during Huawei CSI installation. The default driver name is **csi.huawei.com**.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
reclaimPolicy	Reclamation policy. This parameter is optional. The following types are supported: • **Delete**: Resources are automatically reclaimed. • **Retain**: Resources are manually reclaimed.	• **Delete**: When a PV/PVC is deleted, resources on the storage device are also deleted. • **Retain**: When a PV/PVC is deleted, resources on the storage device are not deleted.
allowVolumeExpansion	Whether to allow volume expansion. This parameter is optional. If this parameter is set to **true**, the capacity of the PV that uses the StorageClass can be expanded.	This function can only be used to expand PV capacity but cannot be used to reduce PV capacity. The PV capacity expansion function is supported in Kubernetes 1.14 (alpha) and later versions.
parameters.backend	Name of the backend where the resource to be created is located. This parameter is optional.	If this parameter is not set, Huawei CSI will randomly select a backend that meets the capacity requirements to create resources. You are advised to specify a backend to ensure that the created resource is located on the expected backend.
parameters.pool	Name of the storage resource pool where the resource to be created is located. This parameter is optional. If this parameter is set, **parameters.backend** must also be specified.	If this parameter is not set, Huawei CSI will randomly select a storage pool that meets the capacity requirements from the selected backend to create resources. You are advised to specify a storage pool to ensure that the created resource is located in the expected storage pool.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
parameters.volumeType	Type of the volume to be created. The following types are supported: • **lun**: A LUN is provisioned on the storage side. • **fs**: A file system is provisioned on the storage side.	--
parameters.allocType	Allocation type of the volume to be created. This parameter is optional. The following types are supported: • **thin**: Not all required space is allocated during creation. Instead, the space is dynamically allocated based on the usage. • **thick**: All required space is allocated during creation.	If this parameter is not set, the default value **thin** is used.
parameters.fsType	Type of a host file system. This parameter is mandatory. The supported types are: • ext2 • ext3 • ext4 • xfs	If this parameter is not set, the default value **ext4** is used. This parameter is valid only when **volumeType** of a StorageClass is set to **lun** and **volumeMode** of a PVC is set to **Filesystem**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
parameters.authClient	IP address of the NFS client that can access the volume. This parameter is mandatory when **volumeType** is set to **fs**. You can enter the client host name (a full domain name is recommended), client IP address, or client IP address segment.	The asterisk (*) can be used to indicate any client. If you are not sure about the IP address of the access client, you are advised to use the asterisk (*) to prevent the client access from being rejected by the storage system. If the client host name is used, you are advised to use the full domain name. The IP addresses can be IPv4 addresses, IPv6 addresses, or a combination of IPv4 and IPv6 addresses. You can enter multiple host names, IP addresses, or IP address segments and separate them with semicolons (;) or spaces or by pressing **Enter**. Example: **192.168.0.10;192.168.0.0/24;myserver1.test**
parameters.cloneSpeed	Cloning speed. This parameter is optional. The value ranges from 1 to 4.	If this parameter is not set, the default value **3** is used. **4** indicates the highest speed. This parameter is available when you clone a PVC or create a PVC using a snapshot. For details, see **7.1.3 Cloning a PVC** or **7.1.4 Creating a PVC Using a Snapshot**.
parameters.applicationType	Application type name for creating a LUN or NAS when the backend is OceanStor Dorado V6.	• 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.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
parameters.qos	LUN/NAS QoS settings of the PV on the storage side. This parameter is optional. The value of the parameter is JSON character strings in dictionary format. A character string is enclosed by single quotation marks and the dictionary key by double quotation marks. Example: **'{"maxMBPS": 999, "maxIOPS": 999}'**	For details about the supported QoS configurations, see **Table 7-2**.
parameters.storageQuota	Quota of a PV on the storage device. This parameter is optional and is valid only when NAS is used for connecting to OceanStor Pacific series storage. The value of the parameter is JSON character strings in dictionary format. A character string is enclosed by single quotation marks and the dictionary key by double quotation marks. Example: **'{"spaceQuota": "softQuota", "gracePeriod": 100}'**	For details about the supported quota configurations, see **Table 7-3**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
parameters.hyperMetro	Whether a HyperMetro volume is to be created. This parameter is conditionally optional and needs to be configured when the backend is of the HyperMetro type. • **"true"**: The created volume is a HyperMetro volume. • **"false"**: The created volume is a common volume.	Set this parameter when the used backend is a HyperMetro backend and a HyperMetro volume needs to be provisioned. The default value is **"false"**.
parameters.fsPermission	Permission on the directory mounted to a container. This parameter is optional.	For details about the configuration format, refer to the Linux permission settings, for example, 777 and 755. All SAN storage devices are supported. Only the following NAS storage devices are supported: OceanStor Dorado V6, OceanStor V6, and OceanStor Pacific series 8.1.2 and later versions.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
parameters.rootSquash	Controls the **root** permission of the client. This parameter is optional. The value can be: • **root_squash**: The client cannot access the storage system as user **root**. If a client accesses the storage system as user **root**, the client will be mapped as an anonymous user. • **no_root_squash**: A client can access the storage system as user **root** and has the permission of user **root**.	Only NAS storage is supported.
parameters.allSquash	Whether to retain the user ID (UID) and group ID (GID) of a shared directory. This parameter is optional. The value can be: • **all_squash**: The UID and GID of the shared directory are mapped to anonymous users. • **no_all_squash**: The UID and GID of the shared directory are retained.	Only NAS storage is supported.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
parameters.snapshotDirectoryVisibility	Whether the snapshot directory is visible. This parameter is optional. The value can be: • **visible**: The snapshot directory is visible. • **invisible**: The snapshot directory is invisible.	Only NAS storage is supported.
mountOptions.nfsvers	NFS mount option on the host. The following mount option is supported: **nfsvers**: protocol version for NFS mounting. The value can be **3**, **4**, **4.0**, or **4.1**.	This parameter is optional after the **-o** parameter when the **mount** command is executed on the host. The value is in list format. If the NFS version is specified for mounting, NFS 3, 4.0, and 4.1 protocols are supported (the protocol must be supported and enabled on storage devices). If **nfsvers** is set to **4**, the latest protocol version NFS 4 may be used for mounting due to different OS configurations, for example, 4.1. If protocol 4.0 is required, you are advised to set **nfsvers** to **4.0**.
mountOptions.acl	The DPC namespace supports the ACL function. The DPC client supports POSIX ACL, NFSv4 ACL, and NT ACL authentication.	The descriptions of **acl**, **aclonlyposix**, **cnflush**, and **cflush** are for reference only. For details about the parameters, see ***OceanStor Pacific Series Product Documentation*** and choose **Configuration** > **Basic Service Configuration Guide for File** > **Configuring Basic Services (DPC Scenario)** > **Accessing a DPC Share on a Client** > **Step 2**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
mountOptions.aclonlyposix	The DPC namespace supports POSIX ACL, and the DPC client supports POSIX ACL authentication. The following protocols support POSIX ACL: DPC, NFSv3, and HDFS. If NFSv4 ACL or NT ACL is used, the DPC client cannot identify the ACL of this type. As a result, the ACL of this type does not take effect.	If **aclonlyposix** and **acl** are used together, only **acl** takes effect. That is, the namespace supports the ACL function.
mountOptions.cnflush	Asynchronous disk flushing mode. That is, data is not flushed to disks immediately when files in the namespace are closed.	Asynchronous flushing mode: When a file is closed, data in the cache is not flushed to storage media in synchronous mode. Instead, data is written from the cache to the storage media in asynchronous flushing mode. After the write service is complete, data is flushed from the cache to disks periodically based on the flushing period. In a multi-client scenario, if concurrent operations are performed on the same file, the file size update is affected by the disk flushing period. That is, the file size is updated only after the disk flushing is complete. Generally, the update is completed within several seconds. Synchronous I/Os are not affected by the disk flushing period.
mountOptions.cflush	Synchronous disk flushing mode. That is, data is flushed to disks immediately when files in the namespace are closed.	By default, the synchronous disk flushing mode is used.

 Copyright © Huawei Technologies Co., Ltd.

Table 7-2 Supported QoS configurations

Storage Type	Parameter	Description	Remarks
OceanStor V3/ OceanStor V5	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. This is a restriction policy parameter.	The value is an integer greater than 0, expressed in MB/s.
	MINBANDWIDTH	Minimum bandwidth. This is a protection policy parameter.	The value is an integer greater than 0, expressed in MB/s.
	MAXIOPS	Maximum IOPS. This is a restriction policy parameter.	The value is an integer greater than 0.
	MINIOPS	Minimum IOPS. This is a protection policy parameter.	The value is an integer greater than 0.
	LATENCY	Maximum latency. This is a protection policy parameter.	The value is an integer greater than 0, expressed in ms.
OceanStor Dorado V3	IOTYPE	Read/write type.	The value can be: ● **2**: read and write I/Os
	MAXBANDWIDTH	Maximum bandwidth. This is a restriction policy parameter.	The value is an integer ranging from 1 to 999999999, expressed in MB/s.
	MAXIOPS	Maximum IOPS. This is a restriction policy parameter.	The value is an integer ranging from 100 to 999999999.
OceanStor Dorado V6/ OceanStor V6	IOTYPE	Read/write type.	The value can be: ● **2**: read and write I/Os
	MAXBANDWIDTH	Maximum bandwidth. This is a restriction policy parameter.	The value is an integer ranging from 1 to 999999999, expressed in MB/s.

 Copyright © Huawei Technologies Co., Ltd.

Storage Type	Parameter	Description	Remarks
	MINBANDWIDTH	Minimum bandwidth. This is a protection policy parameter.	The value is an integer ranging from 1 to 999999999, expressed in MB/s.
	MAXIOPS	Maximum IOPS. This is a restriction policy parameter.	The value is an integer ranging from 100 to 999999999.
	MINIOPS	Minimum IOPS. This is a protection policy parameter.	The value is an integer ranging from 100 to 999999999.
	LATENCY	Maximum latency. This is a protection policy parameter.	The value can be **0.5** or **1.5**, expressed in ms.
FusionStorage/ OceanStor Pacific series	maxMBPS	Maximum bandwidth. This is a restriction policy parameter.	This parameter is mandatory. The value is an integer greater than 0, expressed in MB/s.
	maxIOPS	Maximum IOPS. This is a restriction policy parameter.	This parameter is mandatory. The value is an integer greater than 0.

Table 7-3 Supported quota configurations

Parameter	Description	Remarks
spaceQuota	File quota type.	This parameter is mandatory. Only **softQuota** or **hardQuota** can be configured.
gracePeriod	Grace period allowed when the soft quota is configured.	This parameter is conditionally optional only when **spaceQuota** is set to **softQuota**. The value is an integer ranging from 0 to 4294967294.

For details about how to configure a StorageClass in typical scenarios, see the following examples:

- **Setting the Backend and Storage Pool in a StorageClass**

 Copyright © Huawei Technologies Co., Ltd.

- Setting the NFS Access Mode in a StorageClass
- Setting the Local File System Access Mode in a StorageClass
- Setting the DPC Access Mode in a StorageClass
- Setting an Application Type in a StorageClass
- Setting a Soft Quota in a StorageClass
- Setting QoS in a StorageClass
- Setting HyperMetro in a StorageClass
- Setting the Permission on a Mount Directory in a StorageClass

Setting the Backend and Storage Pool in a StorageClass

If multiple Huawei backends are configured in a Kubernetes cluster or a Huawei backend provides multiple storage pools, you are advised to configure the specified backend and storage pool information in the StorageClass. This prevents Huawei CSI from randomly selecting backends and storage pools and ensures that the storage device where the volume resides complies with the plan.

For details about how to set the backend and storage pool for SAN storage, see the following configuration example.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
  backend: "iscsi_dorado_181"
  pool: "pool001"
  volumeType: lun
  allocType: thin
```

For details about how to set the backend and storage pool for NAS storage, see the following configuration example.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
  backend: "iscsi_dorado_181"
  pool: "pool001"
  volumeType: fs
  allocType: thin
  authClient: "*"
```

Setting the NFS Access Mode in a StorageClass

When a container uses an NFS file system as a storage resource, refer to the following configuration example. In this example, NFS version 4.1 is specified for mounting.

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

 Copyright © Huawei Technologies Co., Ltd.

```
  backend: nfs_dorado_181
  pool: pool001
  volumeType: fs
  allocType: thin
  authClient: "192.168.0.10;192.168.0.0/24;myserver1.test"  #use * for all client
mountOptions:
  - nfsvers=4.1
```

Setting the Local File System Access Mode in a StorageClass

If a container uses a LUN of enterprise storage or distributed storage as a storage resource and a file system needs to be formatted as a local file system, refer to the following example.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: iscsi_dorado_181
  pool: pool001
  volumeType: lun
  allocType: thin
  fsType: xfs
```

Setting the DPC Access Mode in a StorageClass

If a container uses OceanStor Pacific series storage and the storage supports DPC-based access, you can configure mounting parameters for DPC-based access in the StorageClass. In this example, **acl** is used as the authentication parameter for mounting, and **cnflush** is used to set the asynchronous disk flushing mode.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: nfs_pacific_101
  pool: pool001
  volumeType: fs
  allocType: thin
  authClient: "*"  #use * for all client
mountOptions:
  - acl
  - cnflush
```

Setting an Application Type in a StorageClass

When a container uses a LUN of OceanStor Dorado V6 as the storage, if the default application type of the storage cannot meet the I/O model requirements of some services (for example, the container provides the database OLAP service), you can configure an application type in the StorageClass to improve storage performance. For details about the application types to be used, see the product documentation of the corresponding storage product.

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

 Copyright © Huawei Technologies Co., Ltd.

```
  backend: iscsi_dorado_181
  pool: pool001
  volumeType: lun
  allocType: thin
  fsType: xfs
  applicationType: Oracle_OLAP
```

Setting a Soft Quota in a StorageClass

If a container uses a file system of OceanStor Pacific series as the storage, you can configure a soft quota in the StorageClass. The following is a configuration example.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: nfs_pacific_101
  pool: pool001
  volumeType: fs
  allocType: thin
  authClient: "*"
  storageQuota: '{"spaceQuota": "softQuota", "gracePeriod": 100}'
mountOptions:
  - nfsvers=3
```

Setting QoS in a StorageClass

When containers use enterprise storage or distributed storage as storage resources, you can set QoS for the storage resources used by containers to ensure that the storage read and write operations of these containers meet certain service levels.

Storage devices of different models or versions support different QoS settings. For details about how to find the configuration items of the corresponding storage devices, see **Table 7-2**. In this example, the backend is OceanStor Dorado V6. For other storage devices, refer to this example.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: iscsi_dorado_181
  pool: pool001
  volumeType: lun
  allocType: thin
  fsType: xfs
  qos: '{"IOTYPE": 2, "MINIOPS": 1000}'
```

After the StorageClass configuration is complete, perform the following steps to create a StorageClass.

Step 1 Run the following command to create a StorageClass based on the .yaml file.

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

Step 2 Run the following command to view the information about the created StorageClass.

```
# kubectl get sc
NAME   PROVISIONER      RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
mysc   csi.huawei.com   Delete          Immediate           false                  34s
```

 Copyright © Huawei Technologies Co., Ltd.

After creating a StorageClass, you can use the StorageClass to create a PV or PVC.

----End

CAUTION

Pay attention to the following when using a StorageClass:

- Do not delete a StorageClass that is being used by a PV. Otherwise, the StorageClass information of the PV will be missing. As a result, an error occurs during mounting on the host.
- Modifications to a StorageClass do not take effect on existing PVs. You need to delete these PVs and create them again using the modified StorageClass to apply the modified parameters.

Setting HyperMetro in a StorageClass

When a container uses an NFS HyperMetro file system as a storage resource, refer to the following configuration example. In this example, the used backend supports HyperMetro, and **hyperMetro** is set to **true**.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: nfs_hypermetro_dorado_181
  pool: pool001
  volumeType: fs
  hyperMetro: "true"
  allocType: thin
  authClient: "*"
```

NOTICE

Before provisioning a NAS HyperMetro volume, you need to configure the HyperMetro relationship between two storage devices, including the remote device, HyperMetro domain, and the like. The HyperMetro domain of the file system can only work in HyperMetro mode. For details about the configuration operation, see the product documentation of the corresponding storage model.

Setting the Permission on a Mount Directory in a StorageClass

To modify the permission on a mount directory in a container, you can configure the directory permission in a StorageClass. The following is a configuration example.

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

 Copyright © Huawei Technologies Co., Ltd.

```
allocType: thin
authClient: "*"
fsPermission: "777"
rootSquash: "no_root_squash"  # This parameter supports only NAS storage.
all_squash: "no_all_squash"   # This parameter supports only NAS storage.
```

7.1.1.1.2 Configuring a PVC

After configuring a StorageClass, you can use the StorageClass to configure a PVC. For details about the PVC configuration template, see example file **pvc-*xxxx*.yaml** in the **examples** directory in Huawei CSI software package.

Table 7-4 Parameters in the pvc-xxxx.yaml file

Parameter	Description	Remarks
metadata.name	User-defined name of a PVC object.	Take Kubernetes v1.22.1 as an example. The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit.
spec.volumeMode	Volume mode. This parameter is optional. When LUN volumes are used, the following types are supported: • **Filesystem**: local file system. • **Block**: raw device.	This parameter takes effect when a PV is mounted. The default value is **Filesystem**. • **Filesystem** indicates that a container accesses a PV using a local file system. The local file system type is specified by the **fsType** field in the specified StorageClass. • **Block** indicates that a PV is accessed in raw volume mode.
spec.storageClassName	Name of the StorageClass object.	Name of the StorageClass object required by services.
spec.resources.requests.storage	Size of the volume to be created. The format is ***Gi and the unit is GiB.	The PVC capacity depends on storage specifications and host specifications. For example, OceanStor Dorado 6.1.2 or OceanStor Pacific series 8.1.0 is connected to CentOS 7. If ext4 file systems are used, see **Table 7-5**. If XFS file systems are used, see **Table 7-6**. If NFS or raw devices are used, the capacity must meet the specifications of the used Huawei storage device model and version. 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.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
spec.accessModes	Access mode of the volume. • **RWO** (ReadWriteOnce): A volume can be mounted to a node in read/write mode. This mode also allows multiple Pods running on the same node to access the volume. • **ROX** (ReadOnlyMany): A volume can be mounted to multiple nodes in read-only mode. • **RWX** (ReadWriteMany): A volume can be mounted to multiple nodes in read/write mode. • **RWOP** (ReadWriteOncePod): A volume can only be mounted to a single Pod in read/write mode. Kubernetes 1.22 and later versions support this feature.	• RWO/ROX/RWOP: supported by all types of volumes. RWOP is supported only by Kubernetes 1.22 and later versions. Check whether this feature is enabled for your Kubernetes cluster by referring to **9.7 Enabling the ReadWriteOncePod Feature Gate**. • RWX: supported by volumes whose **volumeMode** is set to **Block** or **NFS**.

Table 7-5 ext4 capacity specifications

Storage Type	Storage Specifications	ext4 Specifications	CSI Specifications
OceanStor Dorado 6.1.2	512 Ki to 256 Ti	50 Ti	512 Ki to 50 Ti
OceanStor Pacific series 8.1.0	64 Mi to 512 Ti	50 Ti	64 Mi to 50 Ti

 Copyright © Huawei Technologies Co., Ltd.

Table 7-6 XFS capacity specifications

Storage Type	Storage Specifications	XFS Specifications	CSI Specifications
OceanStor Dorado 6.1.2	512 Ki to 256 Ti	500 Ti	512 Ki to 500 Ti
OceanStor Pacific series 8.1.0	64 Mi to 512 Ti	500 Ti	64 Mi to 500 Ti

Step 1 Based on service requirements, modify specific parameters by referring to the description in this section and the PVC configuration file example to generate the PVC configuration file to be created, for example, the **mypvc.yaml** file in this example.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  storageClassName: mysc
  resources:
    requests:
      storage: 100Gi
```

Step 2 Run the following command to create a PVC using the configuration file.

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

Step 3 After a period of time, run the following command to view the information about the created PVC. If the PVC status is **Bound**, the PVC has been created and can be used by a Pod.

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

CAUTION

- After the PVC is created, if the PVC is in the **Pending** state after a long time (for example, one minute), refer to **10.6 When a PVC Is Created, the PVC Is in the Pending State**.
- You are advised to create or delete a maximum of 100 PVCs in a batch.

----End

After a PVC is created, you can use the PVC to create a Pod. The following is a simple example of using a PVC. In this example, the created Pod uses the newly created *mypvc*.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
```

 Copyright © Huawei Technologies Co., Ltd.

```
selector:
  matchLabels:
    app: nginx
replicas: 2
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - image: nginx:alpine
      name: container-0
      volumeMounts:
      - mountPath: /tmp
        name: pvc-mypvc
    restartPolicy: Always
    volumes:
    - name: pvc-mypvc
      persistentVolumeClaim:
        claimName:  mypvc                # name of PVC
```

7.1.1.2 Static Volume Provisioning

Static volume provisioning allows administrators to use a resource created on the storage side as a PV for containers in the cluster.

To implement static volume provisioning, perform the following steps:

- Configuring a PV
- Configuring a PVC

7.1.1.2.1 Configuring a PV

If static volume provisioning is used, you do not need to configure a StorageClass. Instead, you can directly create a PV. Before creating a PV, you need to configure a PV. The following example is a configuration file for static volume provisioning.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: mypv
spec:
  volumeMode: Filesystem
  storageClassName: ""
  accessModes:
    - ReadWriteOnce
  csi:
    driver: csi.huawei.com
    volumeHandle: iscsi_dorado_181.lun0001
    fsType: xfs
  capacity:
    storage: 100Gi
```

As shown in the preceding example, in the configuration file for static volume provisioning, **storageClassName** must be set to **""**. Otherwise, Kubernetes will use the default StorageClass. For details about other parameters, see **Table 7-7**.

 Copyright © Huawei Technologies Co., Ltd.

Table 7-7 Static volume provisioning parameters

Parameter	Description	Remarks
metadata.name	User-defined name of a PVC object.	Take Kubernetes v1.22.1 as an example. The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit.
spec.volumeMode	Volume mode. This parameter is optional. When LUN volumes are used, the following types are supported: • **Filesystem**: local file system. • **Block**: raw device.	This parameter takes effect when a PV is mounted. The default value is **Filesystem**. • **Filesystem** indicates that a container accesses a PV using a local file system. The local file system type is specified by the **fsType** field in the specified StorageClass. • **Block** indicates that a PV is accessed in raw volume mode.
spec.storageClassName	Name of the StorageClass object. This parameter is mandatory.	Set the parameter to an empty string, that is, enter **""**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
spec.accessModes	Access mode of the volume. • **RWO** (ReadWriteOnce): A volume can be mounted to a node in read/write mode. This mode also allows multiple Pods running on the same node to access the volume. • **ROX** (ReadOnlyMany): A volume can be mounted to multiple nodes in read-only mode. • **RWX** (ReadWriteMany): A volume can be mounted to multiple nodes in read/write mode. • **RWOP** (ReadWriteOncePod): A volume can only be mounted to a single Pod in read/write mode. Kubernetes 1.22 and later versions support this feature.	• RWO/ROX/RWOP: supported by all types of volumes. RWOP is supported only by Kubernetes 1.22 and later versions. Check whether this feature is enabled for your Kubernetes cluster by referring to **9.7 Enabling the ReadWriteOncePod Feature Gate**. • RWX: supported by volumes whose **volumeMode** is set to **Block** or **NFS**.
spec.csi.driver	CSI driver name.	The value is fixed to **csi.huawei.com**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
spec.csi.volumeHandle	Unique identifier of a storage resource. This parameter is mandatory. Format: *<backendName>*.*<volume-name>*	The value of this parameter consists of the following parts: ● *<backendName>*: indicates the name of the backend where the volume resides. You can run the following command to obtain the configured backend information. **kubectl get configmap huawei-csi-configmap -n *huawei-csi* -o yaml** ● *<volume-name>*: indicates the name of a resource (LUN/file system) on the storage. You can obtain the value from DeviceManager.
spec.csi.fsType	Type of a host file system. This parameter is optional. The supported types are: ● ext2 ● ext3 ● ext4 ● xfs	If this parameter is not set, the default value **ext4** is used. This parameter is available only when **volumeMode** is set to **Filesystem**.
spec.capacity.storage	Volume size.	Ensure that the size is the same as that of the corresponding resource on the storage. Kubernetes will not invoke CSI to check whether the value of this parameter is correct. Therefore, the PV can be successfully created even if its capacity is inconsistent with that of the corresponding resource on the storage.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
mountOptions.nfsvers	NFS mount option on the host. The following mount option is supported: **nfsvers**: protocol version for NFS mounting. The value can be **3**, **4**, **4.0**, or **4.1**.	This parameter is optional after the **-o** parameter when the **mount** command is executed on the host. The value is in list format. If the NFS version is specified for mounting, NFS 3, 4.0, and 4.1 protocols are supported (the protocol must be supported and enabled on storage devices). If **nfsvers** is set to **4**, the latest protocol version NFS 4 may be used for mounting due to different OS configurations, for example, 4.1. If protocol 4.0 is required, you are advised to set **nfsvers** to **4.0**.
mountOptions.acl	The DPC namespace supports the ACL function. The DPC client supports POSIX ACL, NFSv4 ACL, and NT ACL authentication.	The descriptions of **acl**, **aclonlyposix**, **cnflush**, and **cflush** are for reference only. For details about the parameters, see ***OceanStor Pacific Series Product Documentation*** and choose **Configuration** > **Basic Service Configuration Guide for File** > **Configuring Basic Services (DPC Scenario)** > **Accessing a DPC Share on a Client** > **Step 2**.
mountOptions.aclonlyposix	The DPC namespace supports POSIX ACL, and the DPC client supports POSIX ACL authentication. The following protocols support POSIX ACL: DPC, NFSv3, and HDFS. If NFSv4 ACL or NT ACL is used, the DPC client cannot identify the ACL of this type. As a result, the ACL of this type does not take effect.	If **aclonlyposix** and **acl** are used together, only **acl** takes effect. That is, the namespace supports the ACL function.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
mountOptions.cnflush	Asynchronous disk flushing mode. That is, data is not flushed to disks immediately when files in the namespace are closed.	Asynchronous flushing mode: When a file is closed, data in the cache is not flushed to storage media in synchronous mode. Instead, data is written from the cache to the storage media in asynchronous flushing mode. After the write service is complete, data is flushed from the cache to disks periodically based on the flushing period. In a multi-client scenario, if concurrent operations are performed on the same file, the file size update is affected by the disk flushing period. That is, the file size is updated only after the disk flushing is complete. Generally, the update is completed within several seconds. Synchronous I/Os are not affected by the disk flushing period.
mountOptions.cflush	Synchronous disk flushing mode. That is, data is flushed to disks immediately when files in the namespace are closed.	By default, the synchronous disk flushing mode is used.

Prerequisites

- A storage resource, such as a LUN or file system, required by the PV to be created exists on the storage device. If the storage resource is a file system, you also need to create the share and client information of the file system.
- You have configured the PV configuration file by referring to **Table 7-7**.

Procedure

Step 1 Run the following command to create a PV based on the prepared .yaml file.

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

Step 2 After a period of time, run the following command to view the information about the created PV. If the PV status is **Available**, the PV is successfully created.

```
# kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM           STORAGECLASS   REASON   AGE
mypv      100Gi      RWO            Retain           Available                                  4s
```

----End

 Copyright © Huawei Technologies Co., Ltd.

7.1.1.2.2 Configuring a PVC

After a PV is created in static volume provisioning mode, you can create a PVC based on the PV for containers. The following example is a PVC configuration file for static volume provisioning.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 100Gi
  volumeName: mypv
```

As shown in the preceding example, set the **volumeName** parameter in the PVC configuration file to the PV created in static volume provisioning mode. For details about the parameters, see **Table 7-8**.

Table 7-8 PVC parameters

Parameter	Description	Remarks
metadata.name	User-defined name of a PVC object.	Take Kubernetes v1.22.1 as an example. The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
spec.accessModes	Access mode of the volume. • **RWO** (ReadWriteOnce): A volume can be mounted to a node in read/write mode. This mode also allows multiple Pods running on the same node to access the volume. • **ROX** (ReadOnlyMany): A volume can be mounted to multiple nodes in read-only mode. • **RWX** (ReadWriteMany): A volume can be mounted to multiple nodes in read/write mode. • **RWOP** (ReadWriteOncePod): A volume can only be mounted to a single Pod in read/write mode. Kubernetes 1.22 and later versions support this feature.	• RWO/ROX/RWOP: supported by all types of volumes. RWOP is supported only by Kubernetes 1.22 and later versions. Check whether this feature is enabled for your Kubernetes cluster by referring to **9.7 Enabling the ReadWriteOncePod Feature Gate**. • RWX: supported by volumes whose **volumeMode** is set to **Block** or **NFS**.
spec.volumeMode	Volume mode.	This parameter is optional. The value can be **Filesystem** or **Block**. The default value is **Filesystem**. This parameter takes effect when a Pod is created. **Filesystem** indicates that a file system is created on a PVC to access the storage. **Block** indicates that a raw volume is used to access the storage.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
spec.resources.requests.storage	Size of the volume to be created.	Size of the volume to be created. The format is ***Gi and the unit is GiB. The PVC capacity depends on storage specifications and host specifications. For example, OceanStor Dorado 6.1.2 or OceanStor Pacific series 8.1.0 is connected to CentOS 7. If ext4 file systems are used, see **Table 7-5**. If XFS file systems are used, see **Table 7-6**. If NFS or raw devices are used, the capacity must meet the specifications of the used Huawei storage device model and version. 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. When a PVC is created using a static PV and the PVC capacity is smaller than the capacity of the bound PV, the PVC capacity is set to the capacity of the bound PV. If the PVC capacity is greater than the capacity of the bound PV, the PVC cannot be created.
spec.volumeName	Name of the PV object.	This parameter is mandatory when a PVC is created statically.

Procedure

Step 1 Run the following command to create a PVC based on the configured .yaml file.

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

Step 2 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 after a long time (for example, one minute), refer to **10.6 When a PVC Is Created, the PVC Is in the Pending State**.
- You are advised to create or delete a maximum of 100 PVCs in a batch.

----End

 Copyright © Huawei Technologies Co., Ltd.

After a PVC is created, you can use the PVC to create a Pod. The following is a simple example of using a PVC. In this example, the created Pod uses the newly created *mypvc*.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:alpine
        name: container-0
        volumeMounts:
        - mountPath: /tmp
          name: pvc-mypvc
      restartPolicy: Always
      volumes:
      - name: pvc-mypvc
        persistentVolumeClaim:
          claimName:  mypvc                  # name of PVC
```

7.1.2 Expanding the Capacity of a PVC

When the capacity of a PVC used by a container is insufficient, you need to expand the capacity of the PVC.

Prerequisites

- A PVC has been created, the backend to which it resides exists and supports capacity expansion.
- For details about the storage devices that support capacity expansion, see **Table 2-4** and **Table 2-6**.
- The csi-resizer service is enabled for huawei-csi-controller.

  ```
  # kubectl describe deploy huawei-csi-controller -n huawei-csi | grep csi-resizer
    csi-resizer:
     Image:      k8s.gcr.io/sig-storage/csi-resizer:v1.3.0
  ```

Procedure

Step 1 Run the **kubectl get pvc** *mypvc* command to query the StorageClass name of the PVC. In the preceding command, *mypvc* indicates the name of the PVC to be expanded.

```
# kubectl get pvc mypvc
NAME            STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS     AGE
mypvc           Bound    pvc-3383be36-537c-4cb1-8f32-a415fa6ba384   2Gi        RWX
mysc            145m
```

Step 2 Run the **kubectl get sc** *mysc* command to check the StorageClass supports capacity expansion. In the preceding command, *mysc* indicates the name of the StorageClass to be queried.

```
# kubectl get sc mysc
NAME            PROVISIONER      RECLAIMPOLICY   VOLUMEBINDINGMODE
```

 Copyright © Huawei Technologies Co., Ltd.

```
ALLOWVOLUMEEXPANSION  AGE
mysc        csi.huawei.com  Delete    Immediate    true        172m
```

If the value of **ALLOWVOLUMEEXPANSION** is **true**, the current StorageClass supports capacity expansion. In this case, go to **Step 4**.

Step 3 Run the following command to change the value of **allowVolumeExpansion** to **true**. In the preceding command, *mysc* indicates the name of the StorageClass to be modified.

```
# kubectl patch sc mysc --patch '{"allowVolumeExpansion":true}'
```

Step 4 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. For example, OceanStor Dorado 6.1.2 or OceanStor Pacific series 8.1.0 is connected to CentOS 7. If ext4 file systems are used, see **Table 7-5**. If XFS file systems are used, see **Table 7-6**. If NFS or raw devices are used, the capacity must meet the specifications of the used Huawei storage device model and version.
- 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.
- If the capacity expansion fails because the target capacity exceeds the storage pool capacity, see **10.17 Failed to Expand the PVC Capacity Because the Target Capacity Exceeds the Storage Pool Capacity**.

Step 5 Run the following command to check whether the capacity modification takes effect.

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

----End

7.1.3 Cloning a PVC

This section describes how to clone a PVC.

When cloning a PVC, you need to specify the data source. The following is a simple example of cloning a PVC. In this example, **mypvc** is used as the data source and a PVC named **myclone** is created.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: myclone
spec:
  storageClassName: mysc
  dataSource:
    name: mypvc
    kind: PersistentVolumeClaim
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
```

CAUTION

- The specified **storageClassName** must be the same as the StorageClass of the source volume in **dataSource**.
- The capacity of the clone volume must be greater than or equal to that of the source volume. Equal capacity is recommended.

Prerequisites

The source PVC already exists in the system, and the backend where the source PVC resides supports cloning. For details about the storage devices that support cloning, see **Table 2-4** and **Table 2-6**.

Procedure

Step 1 Run the following command to create a PVC based on the configuration file of the clone volume.

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

----End

7.1.4 Creating a PVC Using a Snapshot

This section describes how to create a PVC using a snapshot.

When creating a PVC, you need to specify the data source. The following is a simple example of creating a PVC using a snapshot. In this example, **mysnapshot** is used as the data source and a PVC named **myrestore** is created.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myrestore
spec:
  storageClassName: mysc
  dataSource:
    name: mysnapshot
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

CAUTION

- The specified **storageClassName** must be the same as the StorageClass of the snapshot source volume in **dataSource**.
- The capacity of the clone volume must be greater than or equal to that of the snapshot. Equal capacity is recommended.

Copyright © Huawei Technologies Co., Ltd.

Prerequisites

A snapshot already exists in the system, and the backend where the snapshot resides supports cloning. For details about the storage devices that support PVC creation using a snapshot, see **Table 2-4** and **Table 2-6**.

Procedure

Step 1 Run the following command to create a PVC based on the configuration file for creating a volume using a snapshot.

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

----End

7.2 Creating a VolumeSnapshot

In Kubernetes, a **VolumeSnapshot** is a snapshot of a volume on a storage system. The VolumeSnapshot capability provides Kubernetes users with a standard way to replicate the content of a volume at a specified point in time without creating a volume. For example, this function enables database administrators to back up the database before making changes such as editing or deleting.

This section describes how to create a VolumeSnapshot using Huawei CSI. To create a VolumeSnapshot, perform the following steps:

- Configuring a VolumeSnapshotClass
- Configuring a VolumeSnapshot

7.2.1 Configuring a VolumeSnapshotClass

VolumeSnapshotClass provides a way to describe the "classes" of storage when provisioning a VolumeSnapshot. Each VolumeSnapshotClass contains the **driver**, **deletionPolicy**, and **parameters** fields, which are used when a VolumeSnapshot belonging to the class needs to be dynamically provisioned.

The name of a VolumeSnapshotClass object is significant, and is how users can request a particular class. Administrators set the name and other parameters of a class when first creating VolumeSnapshotClass objects, and the objects cannot be updated once they are created.

The following is an example of a VolumeSnapshotClass used by Huawei CSI:

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

You can modify the parameters according to **Table 7-9**. Currently, Huawei CSI does not support user-defined parameters (**parameters**) in a VolumeSnapshotClass. Therefore, you are advised to create a VolumeSnapshotClass for all snapshots.

Copyright © Huawei Technologies Co., Ltd.

Table 7-9 VolumeSnapshotClass parameters

Parameter	Description	Remarks
metadata.name	User-defined name of a VolumeSnapshotClass object.	Take Kubernetes v1.22.1 as an example. The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit.
driver	driver identifier. This parameter is mandatory.	Set this parameter to the driver name set during Huawei CSI installation. The default driver name is **csi.huawei.com**.
deletionPolicy	Snapshot deletion policy. This parameter is mandatory. The value can be: • Delete • Retain	• If the deletion policy is **Delete**, the snapshot on the storage device will be deleted together with the VolumeSnapshotContent object. • If the deletion policy is **Retain**, the snapshot and VolumeSnapshotContent object on the storage device will be retained.

Prerequisites

Huawei CSI supports snapshots, and the volume snapshot component CRD on which its running depends has been installed. For details about the CRD, see **3.5 Checking Volume Snapshot-Dependent Components**.

Procedure

Step 1 Run the following command to create a VolumeSnapshotClass using the created VolumeSnapshotClass configuration file.

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

Step 2 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

7.2.2 Configuring a VolumeSnapshot

VolumeSnapshot can be provisioned in two ways: pre-provisioning and dynamic provisioning. Currently, Huawei CSI supports only dynamic provisioning. This section describes how to dynamically provision a VolumeSnapshot using Huawei CSI.

The following is an example of the VolumeSnapshot configuration file:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: mysnapshot
```

 Copyright © Huawei Technologies Co., Ltd.

```
spec:
  volumeSnapshotClassName: mysnapclass
  source:
    persistentVolumeClaimName: mypvc
```

You can modify the parameters according to **Table 7-10**.

Table 7-10 VolumeSnapshot parameters

Parameter	Description	Remarks
metadata.name	User-defined name of a VolumeSnapshot object.	Take Kubernetes v1.22.1 as an example. The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit.
spec.volumeSnapshotClassName	Name of the VolumeSnapshotClass object.	--
spec.source.persistentVolumeClaimName	Name of the source PVC object.	Name of the source PVC of the snapshot

Prerequisites

- The source PVC exists, and the backend where the PVC resides supports VolumeSnapshot creation. For details about the storage devices that support VolumeSnapshot creation, see **Table 2-4** and **Table 2-6**.
- The volume snapshot component CRD on which the running of Huawei CSI depends has been installed. For details, see **3.5 Checking Volume Snapshot-Dependent Components**.
- A VolumeSnapshotClass that uses Huawei CSI exists in the system.

Procedure

Step 1 Run the following command to create a VolumeSnapshot using the created VolumeSnapshot configuration file.

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

Step 2 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

Copyright © Huawei Technologies Co., Ltd.

8 Advanced Features

8.1 Configuring ALUA

8.2 Configuring Storage Topology Awareness

8.1 Configuring ALUA

Asymmetric Logical Unit Access (ALUA) is a model that supports access to multiple target ports. In the multipathing state, ALUA presents active/passive volumes to the host and provides a port access status switchover interface to switch over the working controllers for volumes. For example, when a volume of a controller fails, you can set the status of ports on the controller to **Unavailable**. After the host multipathing software that supports ALUA detects the status, it switches subsequent I/Os from the failed controller to the peer controller.

8.1.1 Configuring ALUA Using Helm

8.1.1.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend

For details about how to configure ALUA for Huawei enterprise storage, see the host connectivity guide of the corresponding product.

The ALUA configuration may vary according to the OS. Visit **Huawei Technical Support**, 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. Configure ALUA according to the actual situation and the description in the guide. Huawei CSI will apply the configuration items you set to the initiator of the host on Huawei storage.

NOTE

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.

 Copyright © Huawei Technologies Co., Ltd.

ALUA Parameters for OceanStor V3/V5 and OceanStor Dorado V3 Series

Table 8-1 lists the ALUA parameters supported by Huawei CSI for OceanStor V3/V5 and OceanStor Dorado V3 series.

Table 8-1 ALUA parameters supported by Huawei CSI for OceanStor V3/V5 and OceanStor Dorado V3 series

Parameter	Description	Remarks
HostName	Host name rule. This parameter is mandatory. You can use a regular expression.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. When **HostName** is set to *, the configuration takes effect on hosts with any name. For details, see **Regular expression**. If the host name of a compute node matches multiple ALUA configuration options, they will be sorted based on the matching accuracy and the first ALUA configuration option will be used. For details about the sorting rules, see **Rules for Matching ALUA Configuration Items with Host Names**.
MULTIPATHTYPE	Multipathing type. This parameter is mandatory. The value can be: • **0**: Third-party multipathing is not used. • **1**: Third-party multipathing is used.	--
FAILOVERMODE	Initiator switchover mode. This parameter is conditionally mandatory. 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. Configure the initiator switchover mode by referring to the connectivity guide.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
SPECIALMODETYPE	Special mode type of the initiator. This parameter is conditionally mandatory. 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. Configure the special mode type of the initiator by referring to the connectivity guide.
PATHTYPE	Initiator path type. This parameter is conditionally mandatory. The value can be: • **0**: preferred path • **1**: non-preferred path	This parameter needs to be specified only when third-party multipathing is used. Configure the initiator path type by referring to the connectivity guide.

The following uses OceanStor 18500 V5 as an example to describe how to connect to Red Hat. For details about the host connectivity guide, see ***Huawei SAN Storage Host Connectivity Guide for Red Hat***.

The following ALUA configuration example is recommended in the OceanStor 18500 V5 host connectivity guide for Red Hat in non-HyperMetro storage scenarios. In this example, the OS on compute node **myhost01** in the Kubernetes cluster is RHEL 5.*x*, and that on other compute nodes is RHEL 7.*x*. According to the recommendation, the switchover mode of RHEL 5.*x* should be "ALUA not used", and that of RHEL 7.*x* should be "common ALUA".

```
backends:
  - storage: "oceanstor-san"
    name: "oceanstor-iscsi-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "iscsi"
      portals:
        - "192.168.128.120"
        - "192.168.128.121"
      ALUA:
        ^myhost01$:
          MULTIPATHTYPE: 1
          FAILOVERMODE: 2
          PATHTYPE: 0
        *:
          MULTIPATHTYPE: 1
          FAILOVERMODE: 1
          PATHTYPE: 0
```

 Copyright © Huawei Technologies Co., Ltd.

ALUA Parameters for OceanStor V6 and OceanStor Dorado V6 Series

Table 8-2 lists the ALUA parameters supported by Huawei CSI for OceanStor V6 and OceanStor Dorado V6 series.

NOTE

By default, the initiator host access mode of OceanStor V6 and OceanStor Dorado V6 series storage is "balanced mode". Therefore, you are not advised to configure ALUA parameters for OceanStor V6 and OceanStor Dorado V6 series storage.

Table 8-2 ALUA parameters for OceanStor V6 and OceanStor Dorado V6 series

Parameter	Description	Remarks
HostName	Host name rule. This parameter is mandatory. You can use a regular expression.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. When **HostName** is set to *, the configuration takes effect on hosts with any name. For details, see **Regular expression**. If the host name of a compute node matches multiple ALUA configuration options, they will be sorted based on the matching accuracy and the first ALUA configuration option will be used. For details about the sorting rules, see **Rules for Matching ALUA Configuration Items with Host Names**.
accessMode	Host access mode. This parameter is mandatory. The value can be: • **0**: balanced mode • **1**: asymmetric mode	The balanced mode is recommended in non-HyperMetro scenarios. Currently, Huawei CSI does not support SAN HyperMetro scenarios. Exercise caution when using the 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. Currently, Huawei CSI does not support SAN HyperMetro scenarios. Exercise caution when using the asymmetric mode.

 Copyright © Huawei Technologies Co., Ltd.

The following uses OceanStor Dorado 18000 V6 as an example to describe how to connect to Red Hat. For details about the host connectivity guide, see ***OceanStor Dorado 6.x and OceanStor 6.x Host Connectivity Guide for Red Hat***.

The following ALUA configuration example is recommended in the OceanStor Dorado 18000 V6 host connectivity guide for Red Hat in non-HyperMetro storage scenarios.

```
backends:
  - storage: "oceanstor-san"
    name: "dorado-iscsi-155"
    urls:
      - "https://192.168.129.155:8088"
      - "https://192.168.129.156:8088"
    pools:
      - "StoragePool001"
    parameters:
      protocol: "iscsi"
      portals:
        - "192.168.128.120"
        - "192.168.128.121"
      ALUA:
        *:
          accessMode: 0
```

Rules for Matching ALUA Configuration Items with Host Names

- If the configured host name rule exactly matches the host name of the service node, the ALUA configuration item corresponding to the host name rule is used.

  For example, the host name rule in configuration item 1 is ***** and that in configuration item 2 is **^myhost01$**. If the host name of a compute node is **myhost01**, it exactly matches configuration item 2. In this case, Huawei CSI will apply the configuration information in configuration item 2 to the storage side.

- If the configured host name rule does not exactly match the host name of the service node, the first ALUA configuration item matched by regular expressions is used.

  For example, the host name rule in configuration item 1 is **myhost0[0-9]** and that in configuration item 2 is **myhost0[5-9]**. In this case, configuration item 1 has a higher priority than configuration item 2. If the host name of a compute node is **myhost06**, both configuration items can be matched. In this case, Huawei CSI will apply the configuration information in configuration item 1 to the storage side.

8.1.1.2 Configuring ALUA Parameters for a Distributed Storage Backend

For details about how to configure ALUA for Huawei distributed storage, see the host connectivity guide of the corresponding product.

The ALUA configuration may vary according to the OS. Visit **Huawei Technical Support**, 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. Configure ALUA according to the actual situation and the description in the guide. Huawei CSI will apply the configuration items you set to the initiator of the host on Huawei storage.

 Copyright © Huawei Technologies Co., Ltd.

NOTE

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.

In non-HyperMetro scenarios of distributed storage, you are advised to set the switchover mode to "disable ALUA" (default value). This is because the storage system is in active/active mode and "enables ALUA" is meaningless. Therefore, you are not advised to configure ALUA parameters for distributed storage.

Table 8-3 lists the ALUA parameters supported by Huawei CSI for distributed storage.

Table 8-3 ALUA parameters for distributed storage

Parameter	Description	Remarks
HostName	The value of **HostName** is the host name of a worker node, for example, **HostName1** and **HostName2**.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. When **HostName** is set to *, the configuration takes effect on hosts with any name. For details, see **Regular expression**. If the host name of a compute node matches multiple ALUA configuration options, they will be sorted based on the matching accuracy and the first ALUA configuration option will be used. For details about the sorting rules, see **Rules for Matching ALUA Configuration Items with Host Names**.
switchoverMode	Switchover mode. This parameter is mandatory. The value can be: • **Disable_alua**: disables ALUA. • **Enable_alua**: enables ALUA.	In non-HyperMetro scenario, you are advised to set the switchover mode to "disable ALUA". This is because the storage system is in active/active mode and "enables ALUA" is meaningless. Currently, Huawei CSI does not support SAN HyperMetro scenarios. Exercise caution when enabling ALUA.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
pathType	Path type. This parameter is conditionally mandatory. The value can be: • **optimal_path**: preferred path • **non_optimal_path**: non-preferred path	This parameter is mandatory when the switchover mode is set to "enables ALUA".

Rules for Matching ALUA Configuration Items with Host Names

- If the configured host name rule exactly matches the host name of the service node, the ALUA configuration item corresponding to the host name rule is used.

  For example, the host name rule in configuration item 1 is ***** and that in configuration item 2 is **^myhost01$**. If the host name of a compute node is **myhost01**, it exactly matches configuration item 2. In this case, Huawei CSI will apply the configuration information in configuration item 2 to the storage side.
- If the configured host name rule does not exactly match the host name of the service node, the first ALUA configuration item matched by regular expressions is used.

  For example, the host name rule in configuration item 1 is **myhost0[0-9]** and that in configuration item 2 is **myhost0[5-9]**. In this case, configuration item 1 has a higher priority than configuration item 2. If the host name of a compute node is **myhost06**, both configuration items can be matched. In this case, Huawei CSI will apply the configuration information in configuration item 1 to the storage side.

8.1.2 Manually Configuring ALUA

8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend

For details about how to configure ALUA for Huawei enterprise storage, see the host connectivity guide of the corresponding product.

The ALUA configuration may vary according to the OS. Visit **Huawei Technical Support**, 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. Configure ALUA according to the actual situation and the description in the guide. Huawei CSI will apply the configuration items you set to the initiator of the host on Huawei storage.

NOTE

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.

 Copyright © Huawei Technologies Co., Ltd.

ALUA Parameters for OceanStor V3/V5 and OceanStor Dorado V3 Series

Table 8-4 lists the ALUA parameters supported by Huawei CSI for OceanStor V3/V5 and OceanStor Dorado V3 series.

Table 8-4 ALUA parameters supported by Huawei CSI for OceanStor V3/V5 and OceanStor Dorado V3 series

Parameter	Description	Remarks
HostName	Host name rule. This parameter is mandatory. You can use a regular expression.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. When **HostName** is set to *, the configuration takes effect on hosts with any name. For details, see **Regular expression**. If the host name of a compute node matches multiple ALUA configuration options, they will be sorted based on the matching accuracy and the first ALUA configuration option will be used. For details about the sorting rules, see **Rules for Matching ALUA Configuration Items with Host Names**.
MULTIPATHTYPE	Multipathing type. This parameter is mandatory. The value can be: • **0**: Third-party multipathing is not used. • **1**: Third-party multipathing is used.	--
FAILOVERMODE	Initiator switchover mode. This parameter is conditionally mandatory. 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. Configure the initiator switchover mode by referring to the connectivity guide.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
SPECIALMODETYPE	Special mode type of the initiator. This parameter is conditionally mandatory. 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. Configure the special mode type of the initiator by referring to the connectivity guide.
PATHTYPE	Initiator path type. This parameter is conditionally mandatory. The value can be: • **0**: preferred path • **1**: non-preferred path	This parameter needs to be specified only when third-party multipathing is used. Configure the initiator path type by referring to the connectivity guide.

The following uses OceanStor 18500 V5 as an example to describe how to connect to Red Hat. For details about the host connectivity guide, see ***Huawei SAN Storage Host Connectivity Guide for Red Hat***.

The following ALUA configuration example is recommended in the OceanStor 18500 V5 host connectivity guide for Red Hat in non-HyperMetro storage scenarios. In this example, the OS on compute node **myhost01** in the Kubernetes cluster is RHEL 5.*x*, and that on other compute nodes is RHEL 7.*x*. According to the recommendation, the switchover mode of RHEL 5.*x* should be "ALUA not used", and that of RHEL 7.*x* should be "common ALUA".

```
{
  "backends": [
  {
    "storage": "oceanstor-san",
    ...
    "parameters": {..., "ALUA": {"^myhost01$": {"MULTIPATHTYPE": 1, "FAILOVERMODE": 2,
"PATHTYPE": 0}, "*": {"MULTIPATHTYPE": 1, "FAILOVERMODE": 1, "PATHTYPE": 0}}}
  }
  ]
}
```

ALUA Parameters for OceanStor V6 and OceanStor Dorado V6 Series

Table 8-5 lists the ALUA parameters supported by Huawei CSI for OceanStor V6 and OceanStor Dorado V6 series.

NOTE

By default, the initiator host access mode of OceanStor V6 and OceanStor Dorado V6 series storage is "balanced mode". Therefore, you are not advised to configure ALUA parameters for OceanStor V6 and OceanStor Dorado V6 series storage.

 Copyright © Huawei Technologies Co., Ltd.

Table 8-5 ALUA parameters for OceanStor V6 and OceanStor Dorado V6 series

Parameter	Description	Remarks
HostName	Host name rule. This parameter is mandatory. You can use a regular expression.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. When **HostName** is set to *, the configuration takes effect on hosts with any name. For details, see **Regular expression**. If the host name of a compute node matches multiple ALUA configuration options, they will be sorted based on the matching accuracy and the first ALUA configuration option will be used. For details about the sorting rules, see **Rules for Matching ALUA Configuration Items with Host Names**.
accessMode	Host access mode. This parameter is mandatory. The value can be: • **0**: balanced mode • **1**: asymmetric mode	The balanced mode is recommended in non-HyperMetro scenarios. Currently, Huawei CSI does not support SAN HyperMetro scenarios. Exercise caution when using the 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. Currently, Huawei CSI does not support SAN HyperMetro scenarios. Exercise caution when using the asymmetric mode.

The following uses OceanStor Dorado 18000 V6 as an example to describe how to connect to Red Hat. For details about the host connectivity guide, see ***OceanStor Dorado 6.x and OceanStor 6.x Host Connectivity Guide for Red Hat***.

The following ALUA configuration example is recommended in the OceanStor Dorado 18000 V6 host connectivity guide for Red Hat in non-HyperMetro storage scenarios.

```
{
  "backends": [
  {
    "storage": "oceanstor-san",
    ...
```

 Copyright © Huawei Technologies Co., Ltd.

```
    "parameters": {..., "ALUA": {"*": {"accessMode": 0}}}
  }
  ]
}
```

Rules for Matching ALUA Configuration Items with Host Names

- If the configured host name rule exactly matches the host name of the service node, the ALUA configuration item corresponding to the host name rule is used.

  For example, the host name rule in configuration item 1 is ***** and that in configuration item 2 is **^myhost01$**. If the host name of a compute node is **myhost01**, it exactly matches configuration item 2. In this case, Huawei CSI will apply the configuration information in configuration item 2 to the storage side.

- If the configured host name rule does not exactly match the host name of the service node, the first ALUA configuration item matched by regular expressions is used.

  For example, the host name rule in configuration item 1 is **myhost0[0-9]** and that in configuration item 2 is **myhost0[5-9]**. In this case, configuration item 1 has a higher priority than configuration item 2. If the host name of a compute node is **myhost06**, both configuration items can be matched. In this case, Huawei CSI will apply the configuration information in configuration item 1 to the storage side.

8.1.2.2 Configuring ALUA Parameters for a Distributed Storage Backend

For details about how to configure ALUA for Huawei distributed storage, see the host connectivity guide of the corresponding product.

The ALUA configuration may vary according to the OS. Visit **Huawei Technical Support**, 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. Configure ALUA according to the actual situation and the description in the guide. Huawei CSI will apply the configuration items you set to the initiator of the host on Huawei storage.

NOTE

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.

In non-HyperMetro scenarios of distributed storage, you are advised to set the switchover mode to "disable ALUA" (default value). This is because the storage system is in active/active mode and "enables ALUA" is meaningless. Therefore, you are not advised to configure ALUA parameters for distributed storage.

Table 8-6 lists the ALUA parameters supported by Huawei CSI for distributed storage.

 Copyright © Huawei Technologies Co., Ltd.

Table 8-6 ALUA parameters for distributed storage

Parameter	Description	Remarks
HostName	The value of **HostName** is the host name of a worker node, for example, **HostName1** and **HostName2**.	The host name can be obtained by running the **cat /etc/hostname** command. It can be matched by using regular expressions. When **HostName** is set to *, the configuration takes effect on hosts with any name. For details, see **Regular expression**. If the host name of a compute node matches multiple ALUA configuration options, they will be sorted based on the matching accuracy and the first ALUA configuration option will be used. For details about the sorting rules, see **Rules for Matching ALUA Configuration Items with Host Names**.
switchoverMode	Switchover mode. This parameter is mandatory. The value can be: • **Disable_alua**: disables ALUA. • **Enable_alua**: enables ALUA.	In non-HyperMetro scenario, you are advised to set the switchover mode to "disable ALUA". This is because the storage system is in active/active mode and "enables ALUA" is meaningless. Currently, Huawei CSI does not support SAN HyperMetro scenarios. Exercise caution when enabling ALUA.
pathType	Path type. This parameter is conditionally mandatory. The value can be: • **optimal_path**: preferred path • **non_optimal_path**: non-preferred path	This parameter is mandatory when the switchover mode is set to "enables ALUA".

 Copyright © Huawei Technologies Co., Ltd.

Rules for Matching ALUA Configuration Items with Host Names

- If the configured host name rule exactly matches the host name of the service node, the ALUA configuration item corresponding to the host name rule is used.

  For example, the host name rule in configuration item 1 is ***** and that in configuration item 2 is **^myhost01$**. If the host name of a compute node is **myhost01**, it exactly matches configuration item 2. In this case, Huawei CSI will apply the configuration information in configuration item 2 to the storage side.

- If the configured host name rule does not exactly match the host name of the service node, the first ALUA configuration item matched by regular expressions is used.

  For example, the host name rule in configuration item 1 is **myhost0[0-9]** and that in configuration item 2 is **myhost0[5-9]**. In this case, configuration item 1 has a higher priority than configuration item 2. If the host name of a compute node is **myhost06**, both configuration items can be matched. In this case, Huawei CSI will apply the configuration information in configuration item 1 to the storage side.

8.2 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

You need to configure topology labels on worker nodes in the cluster. The method is as follows:

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.
2. 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   v1.22.3
   node02   Ready    worker                 42d   v1.22.3
   node03   Ready    worker                 42d   v1.22.3
   ```

3. 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 8-7**.

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

 Copyright © Huawei Technologies Co., Ltd.

Table 8-7 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.

4. 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"}]
```

8.2.1 Configuring Storage Topology Awareness Using Helm

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

 Copyright © Huawei Technologies Co., Ltd.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the file and configure multiple backends as required. The following is an example. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
backends:
 - storage: "oceanstor-nas"
   name: "storage"
   urls:
     - "https://*.*.*.*:8088"
     - "https://*.*.*.*:8088"
   pools:
     - "***"
     - "***"
   parameters:
     protocol: "nfs"
     portals:
       - "*.*.*.*"
     supportedTopologies:
       - {"topology.kubernetes.io/region": "China-west", "topology.kubernetes.io/zone": "ChengDu"}
       - {"topology.kubernetes.io/region": "China-south","topology.kubernetes.io/zone": "ShenZhen"}
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* ./ **-n** *huawei-csi* **-f** *values.yaml* command to upgrade the Helm chart. The upgrade command will update the **Deployment**, **daemonset**, and **RBAC** resources, but will not update the **secret** resource. If the backend information changes, you must manually update **secret**. For details, see **9.1 Updating the User Name or Password of a Storage Device Configured on CSI**.

In the preceding command, *helm-huawei-csi* indicates the custom chart name and *huawei-csi* indicates the custom namespace.

```
# helm upgrade helm-huawei-csi ./  -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

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 8-8**. 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:
```

 Copyright © Huawei Technologies Co., Ltd.

```
    - 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 8-8 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
```

 Copyright © Huawei Technologies Co., Ltd.

Step 8 Use the StorageClass to create a PVC with the topology capability. For details, see **7.1.1.1.2 Configuring a PVC**.

----End

8.2.2 Manually Configuring Storage Topology Awareness

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.

Multiple backends are separated by commas (,). For details about each backend, see **4.2.1 Creating ConfigMap Files Required for Running Huawei CSI**.

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>** does not need to 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 huawei-csi | 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 huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap              1      5s
```

Step 5 Start huawei-csi services. For details, see **4.2.2 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 8-9**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

 Copyright © Huawei Technologies Co., Ltd.

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 8-9 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.	**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**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description	Remarks
	Both **key** and **value** must be configured in a fixed format.	**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 **7.1.1.1.2 Configuring a PVC**.

----End

 Copyright © Huawei Technologies Co., Ltd.

9 Common Operations

9.1 Updating the User Name or Password of a Storage Device Configured on CSI
9.2 Updating the configmap Object of huawei-csi
9.3 Adding a Backend for huawei-csi
9.4 Updating the huawei-csi-controller Service
9.5 Updating the huawei-csi-node Service
9.6 Modifying the Log Output Mode
9.7 Enabling the ReadWriteOncePod Feature Gate
9.8 Configuring Access to the Kubernetes Cluster as a Non-root User

9.1 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.

```
# chmod +x secretUpdate
```

Step 3 Run the **./secretUpdate -namespace=***huawei-csi* **-logFileDir=***/var/log/huawei/* command. Replace *huawei-csi* with the actual namespace where Huawei CSI runs. If it is not changed, the default value **huawei-csi** is used. Replace */var/log/huawei/* with a directory on which users have the read and write permissions to store Huawei CSI logs. If it is not changed, the default value **/var/log/huawei/** is used. After running the secretUpdate tool, enter the ID of the backend to be configured

 Copyright © Huawei Technologies Co., Ltd.

as prompted. If **Configured** is **false**, the backend is not configured. If **Configured** is **true**, the backend is configured.

```
# ./secretUpdate
Getting backend configuration information.....
-------------------------------------------------------------------------------------
Number  Configured  BackendName        Urls
1       true        strage-backend     [https://192.168.125.25:8088]
2       true        strage-backend-02  [https://192.168.125.26:8088]
3       true        strage-backend-03  [https://192.168.125.27:8088]
4       true        strage-backend-04  [https://192.168.125.28:8088]
5       true        strage-backend-05  [https://192.168.125.29:28443]
6       true        strage-backend-06  [https://192.168.125.30:28443]
-------------------------------------------------------------------------------------
Please enter the backend number to configure (Enter 'exit' to exit):3
```

Step 4 Enter the user name and password as prompted to update the **secret** object.

```
The backend strage-backend-05 has been configured. Update? [Y/N]:y
************************Current backend information**************************
Name:strage-backend-03
Urls:[https://192.168.125.27:8088]
Please enter this backend user name:admin
Please enter this backend password:
Verifying user name and password. Please wait.....................
The acount information of the backend strage-backend-03 has been configured successfully.
```

Step 5 After the configuration is complete, enter **exit** to exit and save the configuration.

```
Please enter the backend number to configure (Enter 'exit' to exit):exit
Saving configuration. Please wait.............
The configuration is saved successfully.
```

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

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

Step 7 Run the following command to restart the huawei-csi-node service.

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

Step 8 Run the **kubectl get pod -A | grep huawei** command to check whether the services are restarted successfully.

```
# kubectl get pod -A | grep huawei
huawei-csi   huawei-csi-controller-695b84b4d8-tg64l   7/7   Running   0   14s
huawei-csi   huawei-csi-node-g6f7z                    3/3   Running   0   14s
```

----End

9.2 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.

9.2.1 Updating the configmap Object Using Helm

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

 Copyright © Huawei Technologies Co., Ltd.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the file, and change the parameter values based on the upgrade requirements. After the change is complete, press **Esc** and enter **:wq!** to save the change. You can modify all parameters provided in **4.1.1 Preparing the values.yaml File**. Exercise caution when modifying backend parameters.

```
# vi values.yaml
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* ./ **-n** *huawei-csi* **-f** *values.yaml* command to upgrade the Helm chart. The upgrade command will update the **Deployment**, **daemonset**, and **RBAC** resources, but will not update the **secret** resource. If the backend information changes, you must manually update **secret**. For details, see **9.1 Updating the User Name or Password of a Storage Device Configured on CSI**.

In the preceding command, *helm-huawei-csi* indicates the custom chart name and *huawei-csi* indicates the custom namespace.

```
# helm upgrade helm-huawei-csi ./  -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

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

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

Step 7 Run the following command to restart the huawei-csi-node service.

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

Step 8 Run the **kubectl get pod -n** *huawei-csi* command to check whether the services are restarted successfully.

```
# kubectl get pod -n huawei-csi
huawei-csi   huawei-csi-controller-695b84b4d8-tg64l   7/7     Running     0          14s
huawei-csi   huawei-csi-node-g6f7z                    3/3     Running     0          14s
```

----End

9.2.2 Manually Updating the configmap Object

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 huawei-csi** command, press **I** or **Insert** to enter the editing mode, and modify related parameters. The iSCSI networking is used as an example. For details about the parameters, see **Table 9-1**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kind: ConfigMap
apiVersion: v1
```

 Copyright © Huawei Technologies Co., Ltd.

```
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
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 9-1 Description of configuration items

Configuration Item	Format	Description	Remarks
data."csi.json".backends	List	List of back-end storage devices to be connected. This parameter is mandatory.	The number of back-end storage devices is not limited. For details about the fields that can be configured for a single back-end storage device, see **Table 9-2**.

Table 9-2 Configuration items of a back-end storage device

Configuration Item	Format	Description	Remarks
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**.
name	String	Storage backend name.	• User-defined character string. The value can contain uppercase letters, lowercase letters, digits, and hyphens (-). • This parameter cannot be modified.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
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 may have multiple management URLs.
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. • Currently, only storage pools can be added. • You can log in to DeviceManager to obtain the storage pools that support the block storage service.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in scenarios where iSCSI is used.	In scenarios where iSCSI 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. Take OceanStor Dorado 6.*x* series as an example. On DeviceManager, choose **Services** > **Network** > **Logical Ports** and obtain the IP address whose data protocol is iSCSI. (For other series, see the corresponding operation description.)

Step 3 If the **storage**, **name**, or **urls** parameter is modified, you need to update the user name or password of the storage device. For details, see **9.1 Updating the User Name or Password of a Storage Device Configured on CSI**.

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

```
# kubectl get deployment huawei-csi-controller -o yaml -n=huawei-csi | 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 -n=huawei-csi | 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
huawei-csi   huawei-csi-controller-695b84b4d8-tg64l   7/7   Running   0   14s
huawei-csi   huawei-csi-node-g6f7z                    3/3   Running   0   14s
```

----End

9.3 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.

Copyright © Huawei Technologies Co., Ltd.

9.3.1 Adding a Backend Using Helm

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the configuration file and add backend configurations under **backends**. The following is an example. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
backends:
  - storage: "oceanstor-san"
    name: "***"
    urls:
      - "https://*.*.*.*:8088"
    pools:
      - "***"
    parameters:
      protocol: "iscsi"
      portals:
        - "*.*.*.*"
  - storage: "oceanstor-nas"
    name: "***"
    urls:
      - "https://*.*.*.*:8088"
    pools:
      - "***"
    parameters:
      protocol: "nfs"
      portals:
        - "*.*.*.*"
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* ./ **-n** *huawei-csi* **-f** *values.yaml* command to upgrade the Helm chart. The upgrade command will update the **Deployment**, **daemonset**, and **RBAC** resources, but will not update the **secret** resource. If the backend information changes, you must manually update **secret**. For details, see **9.1 Updating the User Name or Password of a Storage Device Configured on CSI**.

In the preceding command, *helm-huawei-csi* indicates the custom chart name and *huawei-csi* indicates the custom namespace.

```
# helm upgrade helm-huawei-csi ./  -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----End

Copyright © Huawei Technologies Co., Ltd.

9.3.2 Manually Adding a 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 **4.2.1.12 Connecting to Multiple Backends Using CSI**.

Step 3 Configure accounts for the new backends. For details, see **9.1 Updating the User Name or Password of a Storage Device Configured on CSI**.

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

```
# kubectl get deployment huawei-csi-controller -o yaml -n=huawei-csi | 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 -n=huawei-csi | 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
huawei-csi   huawei-csi-controller-695b84b4d8-tg64l   7/7   Running   0   14s
huawei-csi   huawei-csi-node-g6f7z                    3/3   Running   0   14s
```

----End

9.4 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.

9.4.1 Updating the controller Service Using Helm

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the file and modify controller parameters. The following is an example. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kubernetes:
  namespace: huawei-csi

images:
```

 Copyright © Huawei Technologies Co., Ltd.

```
  # The image name and tag for the attacher, provisioner and registrar sidecars. These must match the
appropriate Kubernetes version.
  sidecar:
    csiAttacher: k8s.gcr.io/sig-storage/csi-attacher:v3.3.0
    csiProvisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
    csiResizer : k8s.gcr.io/sig-storage/csi-resizer:v1.3.0
    livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
    csiSnapshotter: k8s.gcr.io/sig-storage/csi-snapshotter:v4.2.1
    snapshotController: k8s.gcr.io/sig-storage/snapshot-controller:v4.2.1

  # The image name and tag for the Huawei CSI Service container
  # Replace the appropriate tag name
  huaweiCSIService: huawei-csi:3.1.0

# The CSI driver parameter configuration
csi_driver:
  driverName: csi.huawei.com
  endpoint: /csi/csi.sock
  backendUpdateInterval: 60
  controllerLogging:
    module: file
    level: info
    fileDir: /var/log/huawei
    fileSize: 20M
    maxBackups: 9

huaweiCsiController:
  replicas: 1 # Default number of controller replicas

# Default image pull policy for sidecar container images
sidecarImagePullPolicy: "IfNotPresent"

# Default image pull policy for Huawei plugin container images
huaweiImagePullPolicy: "IfNotPresent"

# Flag to enable or disable snapshot (Optional)
snapshot:
  enable: true
# Flag to enable or disable resize (Optional)
resizer:
  enable: true
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* **./ -n** *huawei-csi* command to upgrade the Helm chart.

In the preceding command, *helm-huawei-csi* indicates the name of the chart to be upgraded, **./** indicates the Helm project in the current directory, and *huawei-csi* indicates the namespace where the chart is located.

```
# helm upgrade helm-huawei-csi ./ -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----End

9.4.2 Manually Updating the 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.

 Copyright © Huawei Technologies Co., Ltd.

Step 2 Uninstall the huawei-csi-controller service. For details, see **5.2.2 Uninstalling the huawei-csi-controller Service**.

Step 3 Delete the RBAC permission. For details, see **5.2.5 Deleting the RBAC Permission**.

Step 4 Create the RBAC permission. For details, see **Step 4**.

Step 5 Start the controller service. For details, see **Step 5**.

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 | grep huawei-csi-controller
huawei-csi    huawei-csi-controller-695b84b4d8-tg64l   7/7     Running     0          14s
```

----End

9.5 Updating the huawei-csi-node Service

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

9.5.1 Updating the node Service Using Helm

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the file and modify node parameters. The following is an example. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
kubernetes:
  namespace: huawei-csi

images:
  # The image name and tag for the attacher, provisioner and registrar sidecars. These must match the
appropriate Kubernetes version.
  sidecar:
    livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
    registrar: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0

  # The image name and tag for the Huawei CSI Service container
  # Replace the appropriate tag name
  huaweiCSIService: huawei-csi:3.1.0

# The CSI driver parameter configuration
csi_driver:
  driverName: csi.huawei.com
  endpoint: /csi/csi.sock
  connectorThreads: 4
```


Copyright © Huawei Technologies Co., Ltd.

```
  volumeUseMultipath: true # Flag to enable or disable volume multipath access
  scsiMultipathType: DM-multipath #Required, if volume-use-multipath is set to  TRUE
  nvmeMultipathType: HW-UltraPath-NVMe #Required, if volume-use-multipath is set to  TRUE
  scanVolumeTimeout: 3
  backendUpdateInterval: 60
  nodeLogging:
    module: file
    level: info
    fileDir: /var/log/huawei
    fileSize: 20M
    maxBackups: 9

# Default image pull policy for sidecar container images
sidecarImagePullPolicy: "IfNotPresent"

# Default image pull policy for Huawei plugin container images
huaweiImagePullPolicy: "IfNotPresent"
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* **./ -n** *huawei-csi* command to upgrade the Helm chart.

In the preceding command, *helm-huawei-csi* indicates the name of the chart to be upgraded, **./** indicates the Helm project in the current directory, and *huawei-csi* indicates the namespace where the chart is located.

```
# helm upgrade helm-huawei-csi ./ -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----End

9.5.2 Manually Updating the 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 Uninstall the huawei-csi-node service. For details, see **5.2.1 Uninstalling the huawei-csi-node Service**.

Step 3 Install the new huawei-csi-node service. For details, see **Step 6**.

Step 4 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
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----End

9.6 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**.

 Copyright © Huawei Technologies Co., Ltd.

9.6.1 Modifying the Log Output Mode of the controller or node Service Using Helm

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 Go to the directory where the Helm project is located. If the previous Helm project cannot be found, copy the **helm** directory in the component package to any directory on the master node. For details about the component package path, see **Table 3-1**.

Step 3 Back up the **values.yaml** file used during CSI installation. If the **values.yaml** file used during the last installation cannot be found, run the **helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** command to query the file.

```
# cp values.yaml values.yaml.bak
```

Step 4 Run the **vi** *values.yaml* command to open the file and modify controller or node log parameters. The following is an example. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
# The CSI driver parameter configuration
csi_driver:
  controllerLogging:
    module: file
    level: info
    fileDir: /var/log/huawei
    fileSize: 20M
    maxBackups: 9
  nodeLogging:
    module: file
    level: info
    fileDir: /var/log/huawei
    fileSize: 20M
    maxBackups: 9
```

Step 5 Run the **helm upgrade** *helm-huawei-csi* **./ -n** *huawei-csi* command to upgrade the Helm chart.

In the preceding command, *helm-huawei-csi* indicates the name of the chart to be upgraded, **./** indicates the Helm project in the current directory, and *huawei-csi* indicates the namespace where the chart is located.

```
# helm upgrade helm-huawei-csi ./ -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun  9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----End

9.6.2 Manually 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.

 Copyright © Huawei Technologies Co., Ltd.

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 Uninstall the huawei-csi-controller service. For details, see **5.2.2 Uninstalling the huawei-csi-controller Service**.

Step 3 Go to the **deploy** directory. For details about the component package path, see **Table 3-1**.

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 the following parameters. For details, see **Table 9-3**. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

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

Table 9-3 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 controller service.

```
# kubectl create -f huawei-csi-controller-snapshot-v1.yaml
```

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

Copyright © Huawei Technologies Co., Ltd.

```
# kubectl get pod -A -o wide | grep huawei
huawei-csi   huawei-csi-controller-b59577886-qqzm8   7/7     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 huawei-csi
  ```

----End

9.6.3 Manually 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 Uninstall the huawei-csi-node service. For details, see **5.2.1 Uninstalling the huawei-csi-node Service**.

Step 3 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 **deploy/huawei-csi-node.yaml** in the software package. For details about the parameters, see **Table 9-4**.

```
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 9-4 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**.

Copyright © Huawei Technologies Co., Ltd.

Configuration Item	Description	Remarks
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 4 Run the following command to start the node service.

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

Step 5 After the huawei-csi service is deployed, run the **kubectl get pod -A -o wide | grep huawei-csi-node** command to check whether the service is started.

```
# kubectl get pod -A | grep huawei-csi-node
huawei-csi    huawei-csi-node-4sfwr                3/3     Running            0          18h   10.244.1.68    node
<none>           <none>
```

Step 6 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 huawei-csi
  ```

----End

9.7 Enabling the ReadWriteOncePod Feature Gate

The ReadWriteOnce access mode is the fourth access mode introduced by Kubernetes v1.22 for PVs and PVCs. If you create a Pod using a PVC in ReadWriteOncePod access mode, Kubernetes ensures that the Pod is the only Pod in the cluster that can read or write the PVC.

The ReadWriteOncePod access mode is an alpha feature in Kubernetes v1.22/1.23/1.24. Therefore, you need to enable the ReadWriteOncePod feature in **feature-gates** of kube-apiserver, kube-scheduler, and kubelet before using the access mode.

Procedure

Step 1 Enable the ReadWriteOncePod feature gate for kube-apiserver.

Copyright © Huawei Technologies Co., Ltd.

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.
2. Run the **vi /etc/kubernetes/manifests/kube-apiserver.yaml** command, press **I** or **Insert** to enter the editing mode, and add **--feature-gates=ReadWriteOncePod=true** to the kube-apiserver container. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
...
spec:
  containers:
  - command:
    - kube-apiserver
    - --feature-gates=ReadWriteOncePod=true
    ...
```

NOTE

After the editing is complete, Kubernetes will automatically apply the updates.

Step 2 Enable the ReadWriteOncePod feature gate for kube-scheduler.

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.
2. Run the **vi /etc/kubernetes/manifests/kube-scheduler.yaml** command, press **I** or **Insert** to enter the editing mode, and add **--feature-gates=ReadWriteOncePod=true** to the kube-scheduler container. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
...
spec:
  containers:
  - command:
    - kube-scheduler
    - --feature-gates=ReadWriteOncePod=true
    ...
```

NOTE

After the editing is complete, Kubernetes will automatically apply the updates.

Step 3 Enable the ReadWriteOncePod feature gate for kubelet.

NOTICE

The dynamic Kubelet configuration function is not used since v1.22 and deleted in v1.24. Therefore, you need to perform the following operations on kubelet on each worker node in the cluster.

1. Use a remote access tool, such as PuTTY, to log in to any worker node in the Kubernetes cluster through the management IP address.
2. Run the **vi /var/lib/kubelet/config.yaml** command, press **I** or **Insert** to enter the editing state, and add **ReadWriteOncePod: true** to the **feature-gates** field of the KubeletConfiguration object. If the **feature-gates** field does not exist, add it at the same time. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

```
apiVersion: kubelet.config.k8s.io/v1beta1
featureGates:
  ReadWriteOncePod: true
  ...
```

 Copyright © Huawei Technologies Co., Ltd.

NOTE

The default path of the kubelet configuration file is **/var/lib/kubelet/config.yaml**. Enter the path based on site requirements.

3. After the configuration is complete, run the **systemctl restart kubelet** command to restart kubelet.

----End

9.8 Configuring Access to the Kubernetes Cluster as a Non-root User

Procedure

Step 1 Copy the authentication file of the Kubernetes cluster and modify **/etc/kubernetes/admin.conf** to be the actual authentication file.

```
$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

Step 2 Change the user and user group of the authentication file.

```
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Step 3 Configure the **KUBECONFIG** environment variable of the current user. The following uses Ubuntu 20.04 as an example.

```
$ echo "export KUBECONFIG=$HOME/.kube/config" >> ~/.bashrc
$ source ~/.bashrc
```

----End

 Copyright © Huawei Technologies Co., Ltd.

10 FAQ

10.1 How Do I View Huawei CSI Logs?

10.2 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

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

10.4 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

10.5 Failed to Start huawei-csi Services with the Status Displayed as InvalidImageName

10.6 When a PVC Is Created, the PVC Is in the Pending State

10.7 Before a PVC Is Deleted, the PVC Is in the Pending State

10.8 When a Pod Is Created, the Pod Is in the ContainerCreating State

10.9 A Pod Is in the ContainerCreating State for a Long Time When It Is Being Created

10.10 A Pod Fails to Be Created and the Log Shows That the Execution of the mount Command Times Out

10.11 A Pod Fails to Be Created and the Log Shows That the mount Command Fails to Be Executed

10.12 How Do I Download a Container Image to the Local PC?

10.13 How Do I Obtain CSI Version Information?

10.14 Common Problems and Solutions for Interconnecting with the Tanzu Kubernetes Cluster

10.15 Common Problems and Solutions for Using the Tanzu Kubernetes Cluster

10.16 Failed to Expand the Capacity of a Generic Ephemeral Volume

10.17 Failed to Expand the PVC Capacity Because the Target Capacity Exceeds the Storage Pool Capacity

 Copyright © Huawei Technologies Co., Ltd.

10.1 How Do I View Huawei CSI Logs?

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
huawei-csi   huawei-csi-controller-695b84b4d8-tg64l   7/7     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>_huawei-csi_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
huawei-csi   huawei-csi-node-g6f7z                3/3     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
```

 Copyright © Huawei Technologies Co., Ltd.

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

```
# vi huawei-csi-node-<name>_huawei-csi_huawei-csi-driver-<contrainer-id>.log
```

----End

10.2 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.

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 following 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
```

10.3 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 huawei-csi** command, error message "/var/lib/iscsi is not a directory" is reported.

Root Cause Analysis

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

 Copyright © Huawei Technologies Co., Ltd.

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 file in **Step 6**).

```
# 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 **deploy/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
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----End

10.4 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: 1.18 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.

 Copyright © Huawei Technologies Co., Ltd.

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:

Step 1 Check the residual drive letters on the host.

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 1.2**.
   - If no, no further action is required.

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 1.1**.

   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 1.1**) and perform the clearing operation.
   - If the command execution is suspended, go to **Step 1.3**.
   - If other cases, contact technical support engineers.

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

   a. 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
   ```

   b. Kill the pid.

   ```
   # kill -9 pid
   ```

   c. Record the residual *dm-xx* device and associated disk IDs (for details, see **Step 1.1**) and perform the clearing operation.

Step 2 Clear the residual drive letters on the host.

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

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

 Copyright © Huawei Technologies Co., Ltd.

If an error is reported, contact technical support engineers.

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.

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-3648435a10058805278654321ffffffff -> ../../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-0x648435a10058805278654321ffffffff -> ../../sdb
lrwxrwxrwx 1 root root  9 Aug  2 14:49 wwn-0x68886030000020aff44cc0d060c987f1 -> ../../sdc
```

----End

 Copyright © Huawei Technologies Co., Ltd.

10.5 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
huawei-csi     huawei-csi-controller-fd5f97768-qlldc     6/7     InvalidImageName     0          16s
huawei-csi     huawei-csi-node-25txd                     2/3     InvalidImageName     0          15s
```

Root Cause Analysis

In the .yaml configuration files of the controller and node, the Huawei CSI image version number is incorrect. For example:

```
...
- name: huawei-csi-driver
  image: huawei-csi:3.1.0
...
```

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 -n=huawei-csi
```

NOTE

- In the **image** configuration item under **huawei-csi-driver** in the sample .yaml file, *huawei-csi:*.*.** must be replaced with *<Name>:<Version>* of the created Huawei CSI image.

  ```
  containers:
    ...
    - name: huawei-csi-driver
      image: huawei-csi:3.1.0
  ```

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 -n=huawei-csi
```

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- In the **image** configuration item under **huawei-csi-driver** in the sample .yaml file, *huawei-csi:*.*.** must be replaced with *<Name>:<Version>* of the created Huawei CSI image.

```
containers:
  ...
  - name: huawei-csi-driver
    image: huawei-csi:3.1.0
```

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
huawei-csi   huawei-csi-controller-58799449cf-zvhmv   7/7     Running       0          2m29s
huawei-csi   huawei-csi-node-7fxh6                    3/3     Running       0          12m
```

----End

10.6 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.

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.

 Copyright © Huawei Technologies Co., Ltd.

- 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.
  b. Create a StorageClass. For details, see **7.1.1.1.1 Configuring a StorageClass**.
  c. Create a PVC. For details, see **7.1.1.1.2 Configuring 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.
  b. Delete the StorageClass.
  c. Modify the **StorageClass.yaml** file based on the **Events** information.
  d. Create a StorageClass. For details, see **7.1.1.1.1 Configuring a StorageClass**.
  e. Create a PVC. For details, see **7.1.1.1.2 Configuring 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.

 Copyright © Huawei Technologies Co., Ltd.

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

10.7 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.

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
  ----     ------              ----               ----
```

 Copyright © Huawei Technologies Co., Ltd.

```
-------
  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

10.8 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 **10.1 How Do I View Huawei CSI Logs?**). The error message "Fibre Channel volume device not found" is displayed.

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.

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

10.9 A Pod Is in the ContainerCreating State for a Long Time When It Is Being Created

Symptom

When a Pod is being created, the Pod is in the **ContainerCreating** state for a long time. Check the huawei-csi-node log (for details, see **10.1 How Do I View Huawei CSI Logs?**). No Pod creation information is recorded in the huawei-csi-node log. After the **kubectl get volumeattachment** command is executed, the name of the PV used by the Pod is not displayed in the **PV** column. After a long period of time (more than ten minutes), the Pod is normally created and the Pod status changes to **Running**.

Root Cause Analysis

The kube-controller-manager component of Kubernetes is abnormal.

Solution or Workaround

Contact container platform engineers to rectify the fault.

10.10 A Pod Fails to Be Created and the Log Shows That the Execution of the mount Command Times Out

Symptom

When a Pod is being created, the Pod keeps in the **ContainerCreating** status. In this case, check the log information of huawei-csi-node (for details, see **10.1 How Do I View Huawei CSI Logs?**). The log shows that the execution of the mount command times out.

Root Cause Analysis

1. The possible cause is that the configured service IP address is disconnected. As a result, the **mount** command execution times out and fails.

 Copyright © Huawei Technologies Co., Ltd.

2. For some operating systems, such as Kylin V10 SP2, it takes a long time to run the **mount** command in a container using NFSv3. As a result, the **mount** command may time out.

Solution or Workaround

1. Configure an available service IP address.
2. You are advised to use NFSv4.0 or NFSv4.1.

10.11 A Pod Fails to Be Created and the Log Shows That the mount Command Fails to Be Executed

Symptom

In NAS scenarios, when a Pod is being created, the Pod keeps in the **ContainerCreating** status. In this case, check the log information of huawei-csi-node (for details, see **10.1 How Do I View Huawei CSI Logs?**). The log shows that the mount command fails to be executed.

Root Cause Analysis

The possible cause is that the NFS 4.0/4.1 protocol is not enabled on the storage side. After the NFS v4 protocol fails to be used for mounting, the host does not negotiate to use the NFS v3 protocol for mounting.

Solution or Workaround

- Enable the NFS 3/4/4.0/4.1 protocol on the storage side and retry the default mounting.
- Specify an available NFS protocol for mounting. For details, see **7.1.1.1.1 Configuring a StorageClass**.

10.12 How Do I Download a Container Image to the Local PC?

Download a Container Image Using containerd

Step 1 Run the **ctr image pull** *image:tag* command to download an image to the local PC. In the preceding command, *image:tag* indicates the image to be pulled and its tag.

```
# ctr image pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

Step 2 Run the **ctr image export** *image*.**tar** *image:tag* command to export the image to a file. In the preceding command, *image:tag* indicates the image to be exported, and *image.tar* indicates the name of the exported image file.

```
# ctr image export livenessprobe.tar k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

----End

 Copyright © Huawei Technologies Co., Ltd.

Download a Container Image Using Docker

Step 1 Run the **docker pull** *image:tag* command to download an image to the local PC. In the preceding command, *image:tag* indicates the image to be pulled.

```
# docker pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

Step 2 Run the **docker save** *image:tag* **-o** *image.tar* command to export the image to a file. In the preceding command, *image:tag* indicates the image to be exported, and *image.tar* indicates the name of the exported image file.

```
# docker save k8s.gcr.io/sig-storage/livenessprobe:v2.5.0 -o livenessprobe.tar
```

----End

Download a Container Image Using Podman

Step 1 Run the **podman pull** *image:tag* command to download an image to the local PC. In the preceding command, *image:tag* indicates the image to be pulled.

```
# podman pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

Step 2 Run the **podman save** *image:tag* **-o** *image.tar* command to export the image to a file. In the preceding command, *image:tag* indicates the image to be exported, and *image.tar* indicates the name of the exported image file.

```
# podman save k8s.gcr.io/sig-storage/livenessprobe:v2.5.0 -o livenessprobe.tar
```

----End

10.13 How Do I Obtain CSI Version Information?

This section describes how to view the CSI version.

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 information about the node where huawei-csi-node resides.

```
# kubectl get pod -A  -owide | grep huawei-csi-node
NAMESPACE     NAME                                READY   STATUS    RESTARTS       AGE     IP
NODE          NOMINATED NODE   READINESS GATES
huawei-csi    huawei-csi-node-87mss               3/3     Running   0              6m41s   8.44.128.33
node-1        <none>           <none>
huawei-csi    huawei-csi-node-xp8cc               3/3     Running   0              6m41s   8.44.128.32
node-2        <none>           <none
```

Step 3 Use a remote access tool, such as PuTTY, to log in to any node where huawei-csi-node resides through the node IP address.

Step 4 Run the following command to view the CSI version.

```
# cat /var/lib/kubelet/plugins/csi.huawei.com/version
3.1.0
```

----End

 Copyright © Huawei Technologies Co., Ltd.

10.14 Common Problems and Solutions for Interconnecting with the Tanzu Kubernetes Cluster

This section describes the common problems and solutions for interconnecting with the Tanzu Kubernetes cluster. Currently, the following problems occur during interconnection with the Tanzu Kubernetes cluster:

- A Pod cannot be created because the PSP permission is not created.
- The mount point of the host is different from that of the native Kubernetes. As a result, a volume fails to be mounted.
- The livenessprobe container port conflicts with the Tanzu vSphere port. As a result, the container restarts repeatedly.

10.14.1 A Pod Cannot Be Created Because the PSP Permission Is Not Created

Symptom

When huawei-csi-controller and huawei-csi-node are created, only the Deployment and DaemonSet resources are successfully created, and no Pod is created for the controller and node.

Root Cause Analysis

The service account used for creating resources does not have the "use" permission of the PSP policy.

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 **vi** *psp-use.yaml* command to create a file named **psp-use.yaml**

```
# vi psp-use.yaml
```

Step 3 Configure the **psp-use.yaml** file.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: huawei-csi-psp-role
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs: ['use']
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: huawei-csi-psp-role-cfg
roleRef:
  kind: ClusterRole
  name: huawei-csi-psp-role
  apiGroup: rbac.authorization.k8s.io
subjects:
```

 Copyright © Huawei Technologies Co., Ltd.

```
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: system:serviceaccounts:huawei-csi
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: system:serviceaccounts:default
```

Step 4 Run the **kubectl create -f** *psp-use.yaml* command to create the PSP permission.

```
# kubectl create -f psp-use.yaml
```

----End

10.14.2 Changing the Mount Point of a Host

Symptom

A Pod fails to be created, and error message "mount point does not exist" is recorded in Huawei CSI logs.

Root Cause Analysis

The native Kubernetes cluster in the **pods-dir** directory of huawei-csi-node is inconsistent with the Tanzu Kubernetes cluster.

Solution or Workaround

Step 1 Go to the **helm/esdk/templates** directory and run the **vi huawei-csi-node.yaml** command to open the node configuration file.

```
# vi huawei-csi-node.yaml
```

Step 2 Replace **/var/lib/kubelet/** in the **huawei-csi-node.yaml** file with **/var/vcap/data/kubelet/**. A total of three positions need to be modified.

Position 1:

```
// Before replacement
      - hostPath:
          path: /var/lib/kubelet/plugins_registry
          type: Directory
        name: registration-dir
// After replacement
      - hostPath:
          path: /var/vcap/data/kubelet/plugins_registry
          type: Directory
        name: registration-dir
```

Position 2:

```
// Before replacement
      - hostPath:
          path: /var/lib/kubelet
          type: Directory
        name: pods-dir
// After replacement
      - hostPath:
          path: /var/vcap/data/kubelet
          type: Directory
        name: pods-dir
```

Position 3:

```
// Before replacement
          - mountPath: /var/lib/kubelet
```

 Copyright © Huawei Technologies Co., Ltd.

```
            mountPropagation: Bidirectional
            name: pods-dir
// After replacement
          - mountPath: /var/vcap/data/kubelet
            mountPropagation: Bidirectional
            name: pods-dir
```

----End

10.14.3 Changing the Default Port of the livenessprobe Container

Symptom

The livenessprobe container of the huawei-csi-controller component keeps restarting.

Root Cause Analysis

The default port (9808) of the livenessprobe container of huawei-csi-controller conflicts with the existing vSphere CSI port of Tanzu.

Solution or Workaround

Change the default port of the livenessprobe container to an idle port.

Step 1 Go to the **helm/esdk/templates** directory and run the **vi huawei-csi-controller.yaml** command to open the controller configuration file.

```
# vi huawei-csi-controller.yaml
```

Step 2 Change the default port (9808) of the livenessprobe container to an idle port.

----End

10.15 Common Problems and Solutions for Using the Tanzu Kubernetes Cluster

This section describes the common problems and solutions for using the Tanzu Kubernetes cluster.

10.15.1 Failed to Create an Ephemeral Volume

Symptom

A **generic ephemeral volume** fails to be created, and the error message **PodSecurityPolicy: unable to admit pod: [spec.volumes[0]: Invalid value: "ephemeral": ephemeral volumes are not allowed to be used spec.volumes[0]** is displayed.

Root Cause Analysis

The current PSP policy does not contain the permission to use ephemeral volumes.

 Copyright © Huawei Technologies Co., Ltd.

Solution or Workaround

Add the permission to use ephemeral volumes to the default PSP **pks-privileged** and **pks-restricted**. The following is an example of modifying **pks-privileged**:

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 **kubectl edit psp pks-privileged** to modify the **pks-privileged** configuration.

```
# kubectl edit psp pks-privileged
```

Step 3 Add **ephemeral** to **spec.volumes**. The following is an example.

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  annotations:
    apparmor.security.beta.kubernetes.io/allowedProfileName: '*'
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
  creationTimestamp: "2022-10-11T08:07:00Z"
  name: pks-privileged
  resourceVersion: "1227763"
  uid: 2f39c44a-2ce7-49fd-87ca-2c5dc3bfc0c6
spec:
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  supplementalGroups:
    rule: RunAsAny
  volumes:
  - glusterfs
  - hostPath
  - iscsi
  - nfs
  - persistentVolumeClaim
  - ephemeral
```

Step 4 Run the **kubectl get psp pks-privileged -o yaml** command to check whether the addition is successful.

```
# kubectl get psp pks-privileged -o yaml
```

----End

10.16 Failed to Expand the Capacity of a Generic Ephemeral Volume

Symptom

In an environment where the Kubernetes version is earlier than 1.25, the capacity of a **generic ephemeral volume** of the LUN type fails to be expanded. The system displays a message indicating that the PV capacity has been expanded, but the PVC capacity fails to be updated.

Root Cause Analysis

This problem is caused by a Kubernetes **bug**, which has been resolved in Kubernetes 1.25.

 Copyright © Huawei Technologies Co., Ltd.

10.17 Failed to Expand the PVC Capacity Because the Target Capacity Exceeds the Storage Pool Capacity

Symptom

In a Kubernetes environment earlier than 1.23, PVC capacity expansion fails when the target capacity exceeds the storage pool capacity.

Root Cause Analysis

This is a known issue in the community. For details, see **Recovering from Failure when Expanding Volumes**.

Solution or Workaround

For details, see **Recovering from Failure when Expanding Volumes**.

Copyright © Huawei Technologies Co., Ltd.

11 Appendix

11.1 Example ALUA Configuration Policy of OceanStor V3/V5 and OceanStor Dorado V3

11.2 Example ALUA Configuration Policy of OceanStor Dorado 6.x

11.3 Example ALUA Configuration Policy of Distributed Storage

11.4 Installing Helm 3

11.5 Creating a CSI Image

11.1 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 **8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend**, 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

 Copyright © Huawei Technologies Co., Ltd.

8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend, 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 **8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend**: 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.

11.2 Example ALUA Configuration Policy of OceanStor Dorado 6.*x*

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 **8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend**, 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 **8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend**, 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 **8.1.2.1 Configuring ALUA Parameters for a Huawei Enterprise Storage Backend**: For host **node1**, select

 Copyright © Huawei Technologies Co., Ltd.

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.

11.3 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 **8.1.2.2 Configuring ALUA Parameters for a Distributed Storage Backend**, 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 **8.1.2.2 Configuring ALUA Parameters for a Distributed Storage Backend**, 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 **8.1.2.2 Configuring ALUA Parameters for a Distributed Storage Backend**: 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.

11.4 Installing Helm 3

This section describes how to install Helm 3.

For details, see **https://helm.sh/docs/intro/install/**.

Prerequisites

Ensure that the master node in the Kubernetes cluster can access the Internet.

 Copyright © Huawei Technologies Co., Ltd.

Procedure

Step 1 Run the following command to download the Helm 3 installation script.

```
# curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
```

Step 2 Run the following command to modify the permission on the Helm 3 installation script.

```
# chmod 700 get_helm.sh
```

Step 3 Determine the Helm version to be installed based on the version mapping between Helm and Kubernetes. For details about the version mapping, see **Helm Version Support Policy**. Then run the following command to change the **DESIRED_VERSION** environment variable to the Helm version to be installed and run the installation command.

```
# DESIRED_VERSION=v3.9.0 ./get_helm.sh
```

Step 4 Run the following command to check whether Helm 3 of the specified version is successfully installed.

```
# helm version
version.BuildInfo{Version:"v3.9.0", GitCommit:"7ceeda6c585217a19a1131663d8cd1f7d641b2a7",
GitTreeState:"clean", GoVersion:"go1.17.5"}
```

----End

11.5 Creating a CSI Image

This section describes how to create a CSI image.

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 binary file to the **image** directory.

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

```
# cat <<EOF > ./Dockerfile
FROM busybox:stable-glibc

LABEL maintainers="The Huawei CSI Team"
LABEL description="Huawei Storage CSI Driver."

COPY huawei-csi /

ENTRYPOINT ["/huawei-csi"]

EOF
```

 Copyright © Huawei Technologies Co., Ltd.

NOTICE

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

Step 6 Run the **docker build -f Dockerfile -t huawei-csi:3.1.0 .** command to create an image.

```
# docker build -f Dockerfile -t huawei-csi:3.1.0 .
```

NOTE

3.1.0 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 7 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                    3.1.0               c8b5726118ac        About a minute ago   39 MB
```

Step 8 Run the **docker save huawei-csi:3.1.0 -o huawei-csi.tar** command to export the image.

```
# docker save huawei-csi:3.1.0 -o huawei-csi.tar
```

NOTE

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

----End

 Copyright © Huawei Technologies Co., Ltd.