User Guide
2.2.RC4

Kubernetes CSI

Issue	01
Date	2021-03-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Huawei Technologies Co., Ltd.

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

Website: https://www.huawei.com

Email: support@huawei.com

About This Document

Intended Audience

This document is intended for:

- Technical support engineers
- O&M engineers

Symbol Conventions

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

Symbol	Description
DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Contents

1 Overview

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

2 Environment Support

- Kubernetes has been deployed and is running properly.
- A Huawei storage device is running properly.

Table 2-1 Version mappings between Huawei Enterprise storage and Kubernetes

Kubernetes	Huawei Enterprise storage
1.13/1.14/1.15/1.16/1.17/1.18/1.19	OceanStor Dorado V6 6.0.RC1/6.0.0/6.0.1/6.1.0/6.1.2.RC1 OceanStor Dorado V3 V300R002 OceanStor F V5/V5 V500R007/V500R007 Kunpeng OceanStor F V3/V3 V300R006

Table 2-2 Version mappings between Huawei Distributed storage and Kubernetes

Kubernetes	Huawei Distributed storage
1.13/1.14/1.15/1.16/1.17/1.18/1.19	FusionStorage V100R006C30 FusionStorage Block 8.0.0/8.0.1 OceanStor Pacific 8.1.RC2

Table 2-3 Mappings among Huawei CSI, Features, and Kubernetes versions (√: supported, x: not supported)

Features	1.13	1.14	1.15	1.16	1.17	1.18	1.19
Create PVC	√	√	√	√	√	√	√

Features	1.13	1.14	1.15	1.16	1.17	1.18	1.19
Delete PVC	√	√	√	√	√	√	√
Create Pod	√	√	√	√	√	√	√
Delete Pod	√	√	√	√	√	√	√
Offline Resize	x	x	x	√	√	√	√
Online Resize	x	x	x	√	√	√	√
Create Snapshot	x	x	x	x	√	√	√
Delete Snapshot	x	x	x	x	√	√	√
Restore	x	x	x	x	√	√	√
Clone	x	x	x	x	√	√	√

3 Deployment

NOTICE

- This guide is aimed at containerized deployment methods.
- The plugin itself is still compatible with non-containerized deployment methods, but it is not recommended. Please contact Huawei engineers if necessary. The last version supporting non-containerized deployment is **eSDK Storage Kubernetes Plugins 2.1.0**

3.1 Obtaining the Software Package

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

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

Step 2 Download the package ***eSDK_EnterPrise_Storage_Plugin_.*.**.zip**,*.*.** indicates the release version number.

Step 3 Decompress the package.

Step 4 In the decompressed directory, the package and docs all included.

----End

3.2 Components in the Software Package

Table 3-1 Component description

Component	Description
bin/huawei-csi	Implements the CSI API.
bin/passwdEncrypt	Encrypts plaintext passwords.
yamls	yaml sample file used in subsequent deployment

3.3 Deployment Preparations

3.3.1 Creating a Huawei CSI Image

Step 1 Prepare a host that runs the same operating system as the Kubernetes cluster. Ensure that docker has been installed on the host and the host can access external networks.

Step 2 Create a directory on the host and go to the directory.

```
# mkdir image; cd image
```

Step 3 Copy the huawei-csi component to the current directory.

Step 4 Create file **Dockerfile**.

- Create a CentOS image and edit the image file as follows:

```
FROM centos:***

RUN yum install -y \
    device-mapper-multipath \
    iscsi-initiator-utils \
    xfsprogs \
    nfs-utils \
    net-tools \
    e2fsprogs \
    nvme-cli

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

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

- Create a Ubuntu image and edit the image file as follows:

```
FROM ubuntu:***

RUN apt update && \
    apt -y install \
    multipath-tools \
    open-iscsi \
    nfs-common \
    xfsprogs \
    gawk

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



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

- Create a SUSE image and edit the image file as follows:

```
FROM suse:***

RUN zypper install -y \
    multipath-tools \
    open-iscsi \
    xfsprogs \
    nfs-utils \
    net-tools \
    e2fsprogs \
    libopeniscsiusr0_2_0 \
    nvme-cli

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

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

NOTICE

- Based on the OS where Kubernetes or OpenShift running on, the basic image which Dockerfile bases on may be different. Please contact Huawei engineers if necessary.
- In order to ensure the compatibility of the software in the container and the software on the host, ensure that the container and the host use the same image source or software package for software installation.

Step 5 Make an image.

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

NOTE

.*. indicates the plug-in version number corresponding to the software package name.

Step 6 Export the image.

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

Step 7 Copy the **huawei-csi.tar** image file to all worker nodes in the Kubernetes cluster and import the image.

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

----End

3.3.2 Encrypting a Password

Step 1 Use the passwdEncrypt tool to encrypt the back-end storage management password and record the encrypted password.

```
# ./passwdEncrypt
Enter password: ******
Encrypted password: ******************
```

----End

3.4 Interconnecting with Enterprise Storage

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

3.4.1 Interconnecting with Enterprise Storage SAN over iSCSI

NOTICE

Before deployment, ensure that:

- An iSCSIclient has been installed on all worker nodes.
- All worker nodes can properly connect to the back-end storage management IP address.
- All worker nodes can properly connect to the back-end storage service IP address.
- If multipath networking is used, multipath software has been installed on all worker nodes.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-oceanstor-iscsi.yaml** in the software package.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-san",
                "product": "***",
                "name": "***",
                "urls": [
                    "https://*.*.*.*:8088/deviceManager/rest"
                ],
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"protocol": "iscsi", "portals": ["*.*.*.*"]}
            }
        ]
    }
```


Table 3-2 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
product	String	Type of a storage product.	The value can be **V3**, **V5**, or **Dorado**.
storage	String	Type of the storage backend.	The value is fixed to **oceanstor-san**.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
urls	List	Management URL of the storage backend.	One or more management URLs are supported. Currently, only IPv4 addresses are supported.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.
parameters	Dictionary	Variable parameters in the iSCSI scenario.	In the iSCSI scenario, set **protocol** to **iscsi** and **portals** to the iSCSI service IP address of the storage backend.

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.4.2 Interconnecting with Enterprise Storage SAN over FC

NOTICE

Before deployment, ensure that:

- All worker nodes can properly connect to the back-end storage management IP address.
- All worker nodes can properly connect to the storage backend over FC links.
- If multipath networking is used, multipath software has been installed on all worker nodes.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-oceanstor-fc.yaml** in the software package.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-san",
                "product": "***",
                "name": "***",
                "urls": [
                    "https://*.*.*.*:8088/deviceManager/rest"
                ],
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"protocol": "fc"}
            }
        ]
    }
```

Table 3-3 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
product	String	Type of a storage product.	The value can be **V3**, **V5**, or **Dorado**.
storage	String	Type of the storage backend.	The value is fixed to **oceanstor-san**.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
urls	List	Management URL of the storage backend.	One or more management URLs are supported. Currently, only IPv4 addresses are supported.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.
parameters	Dictionary	Variable parameters in the FC scenario.	In the FC scenario, set **protocol** to **fc**.

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.4.3 Interconnecting with Enterprise Storage NAS over NFS

NOTICE

Before deployment, ensure that:

- An NFS client has been installed on all worker nodes.
- All worker nodes can properly connect to the back-end storage management IP address.
- All worker nodes can properly connect to the IP address of the back-end storage NFS logical port.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-oceanstor-nfs.yaml** in the software package.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-nas",
                "product": "***",
                "name": "***",
                "urls": [
                    "https://*.*.*.*:8088/deviceManager/rest"
                ],
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"portal": "*.*.*.*"}
            }
        ]
    }
```

Table 3-4 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).

Configuration Item	Format	Description	Remarks
product	String	Type of a storage product.	The value can be **V3**, **V5**, or **Dorado**.
storage	String	Type of the storage backend.	The value is fixed to **oceanstor-nas**.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
urls	List	Management URL of the storage backend.	One or more management URLs are supported. Currently, only IPv4 addresses are supported.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.
parameters	Dictionary	Variable parameters in the NAS scenario.	In the NAS scenario, set **portal** to the IP address of the back-end storage NFS logical port or DNS Zone.

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.4.4 Interconnecting with Enterprise Storage NVMe over RoCE

NOTICE

Before deployment, ensure that:

- All worker nodes support NVMe over RoCE.
- The NVMe client has been installed on all worker nodes.
- All worker nodes can properly connect to the back-end storage management IP address.
- All worker nodes can properly connect to the back-end storage service IP address.
- If multipath networking is used, multipath software has been installed on all worker nodes.
- Currently, only Dorado V6 supports NVMe over RoCE.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-oceanstor-roce.yaml** in the software package.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "oceanstor-san",
                "product": "***",
                "name": "***",
                "urls": [
                    "https://*.*.*.*:8088/deviceManager/rest"
                ],
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"protocol": "roce", "portals": ["*.*.*.*", "*.*.*.*"]}
            }
        ]
    }
```

Table 3-5 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
product	String	Type of a storage product.	The value is fixed to **Dorado**.
storage	String	Type of the storage backend.	The value is fixed to **oceanstor-san**.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
urls	List	Management URL of the storage backend.	One or more management URLs are supported. Currently, only IPv4 addresses are supported.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.

Configuration Item	Format	Description	Remarks
parameters	Dictionary	Variable parameters in the RoCE scenario.	In the RoCE scenario, set **protocol** to roce and **portals** to the NVMe over RoCE service IP address of the storage backend.

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.5 Interconnecting with Distributed Storage

This section describes how to interconnect the huawei-csi plug-in with Huawei cloud storage.

3.5.1 Interconnecting with Distributed Storage SAN over SCSI

NOTICE

Before deployment, ensure that:

- All worker nodes can properly connect to the back-end storage management IP address.
- The FusionStorage VBS client has been installed on all worker nodes.
- All worker nodes have been added to the FusionStorage Block client.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-fusionstorage-scsi.yaml** in the software package.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "fusionstorage-san",
                "name": "***",
                "url": "https://*.*.*.*:28443",
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"SCSI": {"***": "*.*.*.*", "***": "*.*.*.*"}}
            }
        ]
    }
```


Table 3-6 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage backend.	The value is fixed to **fusionstorage-san** for interconnection with FusionStorage Block.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
url	String	Management URL of the storage backend.	Management URL of FusionStorage.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.
parameters	Dictionary	Variable parameters.	The parameter format is **{"SCSI": {"hostname":"*.*.*.*"}}**, where **hostname** indicates the host name of a worker node and ***.*.*.*** indicates the managment IP address of the FusionStorage Block client. If there are multiple worker nodes, configure them in dictionary format, separated by commas(,).

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.5.2 Interconnecting with Distributed Storage SAN over iSCSI

NOTICE

Before deployment, ensure that:

- The iSCSI client has been installed on all worker nodes.
- All worker nodes can properly connect to the back-end storage management IP address.
- All worker nodes can properly connect to the back-end storage service IP address.
- If multipath networking is used, multipath software has been installed on all worker nodes.
- Worker nodes name can only contain digits, letters, underscores (_), hyphens (-), periods (.), and colons (:) and starts with a digit, letter, or underscore, and the name length cannot exceed 31 characters.
- Only FusionStorage 8.0 and later versions support iSCSI mode.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-fusionstorage-iscsi.yaml** in the software package.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "fusionstorage-san",
                "name": "***",
                "url": "https://*.*.*.*:28443",
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"ISCSI": ["*.*.*.*"]}
            }
        ]
    }
```

Table 3-7 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).

Configuration Item	Format	Description	Remarks
storage	String	Type of the storage backend.	The value is fixed to **fusionstorage-san** for interconnection with FusionStorage Block.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
url	String	Management URL of the storage backend.	Management URL of FusionStorage.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.
parameters	Dictionary	Variable parameters.	The parameter format is **{"ISCSI": ["*.*.*.*"]}**, where ***.*.*.*** indicates the iSCSI service IP address of the FusionStorage. If there are multiple service IP addresses, configure them in list format.

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.5.3 Interconnecting with Distributed Storage NAS over NFS

NOTICE

Before deployment, ensure that:

- An NFS client has been installed on all worker nodes.
- All worker nodes can properly connect to the back-end storage management IP address.
- All worker nodes can properly connect to the IP address of the back-end storage NFS logical port.

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-configmap.yaml** file. For details, see sample file **yamls/huawei-csi-configmap-fusionstorage-nfs.yaml** in the software package.


```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: kube-system
data:
  csi.json: |
    {
        "backends": [
            {
                "storage": "fusionstorage-nas",
                "name": "***",
                "url": "https://*.*.*.*:28443",
                "user": "***",
                "password": "***",
                "pools": ["***"],
                "parameters": {"portal": "*.*.*.*"}
            }
        ]
    }
```

Table 3-8 Configuration items

Configuration Item	Format	Description	Remarks
name	String	Name of a storage backend.	The value can contain uppercase letters, lowercase letters, digits, and hyphens (-).
storage	String	Type of the storage backend.	The value is fixed to **fusionstorage-nas**.
pools	List	Name of the used storage pool.	One or more storage pools are supported.
url	List	Management URL of the storage backend.	Management URL of FusionStorage.
user	String	Management user name of the storage backend.	
password	String	Management user password of the storage backend.	Use the encrypted password generated in **3.3.2 Encrypting a Password**.
parameters	Dictionary	Variable parameters in the NAS scenario.	In the NAS scenario, set **portal** to the IP address of the back-end storage NFS logical port.

Step 2 Create file **huawei-csi-configmap.yaml**.

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

----End

3.6 Starting the huawei-csi Service

NOTICE

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

Perform the following steps only on any master node.

Step 1 Compile the **huawei-csi-rbac.yaml** file. For details, see sample file **yamls/huawei-csi-rbac.yaml** in the software package. For details of resize and snapshot, please see:

Appendix: Details of th...

Create file **huawei-csi-rbac.yaml**.

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

Step 2 Compile the **huawei-csi-controller.yaml** file. For details, see sample file **yamls/huawei-csi-controller.yaml** in the software package. For details of resize and snapshot, please see: **Appendix: Details of th...**

Start the controller service.

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

NOTE

In the **huawei-csi:*.*.*** field in the yaml file, replace ***.*.*** with the version number of the created Huawei CSI image.

Step 3 Compile the **huawei-csi-node.yaml** file. For details, see sample file **yamls/huawei-csi-node.yaml** in the software package.

Start the node service.

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

NOTE

- In the **huawei-csi:*.*.*** field in the yaml file, replace ***.*.*** with the version number of the created Huawei CSI image.

Step 4 After the preceding steps are complete, the containerized huawei-csi service is deployed.

----End

- **Appendix: Details of the resize and snapshot**

NOTE

The csi-resizer has been supported since Kubernetes v1.16

- For the huawei-csi-rbac.yaml file, please refer to the yamls/huawei-csi-resize-rbac.yaml in the software package
- For the huawei-csi-controller.yaml file, please refer to the yamls/huawei-csi-resize-controller.yaml in the software package

The csi-snapshotter has been supported since Kubernetes v1.17

- For the huawei-csi-rbac.yaml file, please refer to the yamls/huawei-csi-resize-snapshot-rbac.yaml in the software package
- For the huawei-csi-controller.yaml file, please refer to the yamls/huawei-csi-resize-snapshot-controller.yaml in the software package

4 Instructions for Use

This chapter describes how to use Huawei storage to provide PersistentVolume for Kubernetes.

4.1 Creating StorageClass

4.1.1 Creating LUN StorageClass

Configure the StorageClass yaml file.

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

Table 4-1 Parameter description

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

Parameter	Description	Remarks
volumeType	Type of the volume to be created.	The value is fixed to **lun**.
allocType	How the volume is allocated.	This parameter is optional. The value can be **thin** or **thick**, and the default value is **thin**.
cloneFrom	Original volume of the specified clone.	This parameter is optional. The format is *Storage backend name-Original volume name*.
cloneSpeed	Speed of the specified clone.	This parameter is optional. The value ranges from 1 to 4 and the default value is **3**. **4** indicates the highest speed.
fsType	Type of the specified file system.	This parameter is optional. The value can be **ext2**, **ext3**, or **ext4**, and the default value is **ext4**.

Create StorageClass based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```

4.1.2 Creating FileSystem StorageClass

Configure the StorageClass yaml file.

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

Table 4-2 Parameter description

Parameter	Description	Remarks
name	User-defined name of a StorageClass object.	
provisioner	provisioner identifier.	The value is fixed to **csi.huawei.com**.
volumeType	Type of the volume to be created.	The value is fixed to **fs**.

Parameter	Description	Remarks
authClient	Client that can access the FS volume.	This parameter is mandatory. You can enter the client host name, client IP address, or client IP address segment, or use asterisks (*) to represent all client IP addresses. You can specify multiple clients which are separated by semicolons (;).
allocType	How the volume is allocated.	This parameter is optional. The value can be **thin** or **thick**, and the default value is **thin**.
cloneFrom	Original volume of the specified clone.	This parameter is optional. The format is *Storage backend name-Original volume name*.
cloneSpeed	Speed of the specified clone.	This parameter is optional. The value ranges from 1 to 4 and the default value is **3**. **4** indicates the highest speed.

Create StorageClass based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```

4.2 Creating PersistentVolumeClaim

Configure the PersistentVolumeClaim yaml file.

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

Table 4-3 Parameter description

Parameter	Description	Remarks
name	User-defined name of a PersistentVolumeClaim object.	
storageClassName	Name of the StorageClass object.	Set this parameter to the name of the StorageClass object created in **4.1 Creating StorageClass**.

Parameter	Description	Remarks
storage	Size of the volume to be created.	The value format is ***Gi. The unit is GB.
accessModes	The access Mode of the volume	ReadWriteOnce, ReadOnlyMany and ReadWriteMany for LUN. If ReadWriteMany mode is used and there're multiple Pods accessing the LUN simultaneously, the Pod applications are responsible to keep data consistency. ReadWriteOnce, ReadOnlyMany and ReadWriteMany for FileSystem.

Create PersistentVolumeClaim based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```

4.3 Creating Pod

Configure the Pod yaml file.

```
kind: Pod
apiVersion: v1
metadata:
  name: "***"
spec:
  containers:
 - name: "***"
      image: "***"
      volumeMounts:
        - name: mypv
          mountPath: "***"
  volumes:
 - name: mypv
      persistentVolumeClaim:
        claimName: "***"
```

Table 4-4 Parameter description

Parameter	Description	Remarks
metadata:name	User-defined name of a Pod object.	
spec:containers:name	User-defined container name.	
spec:containers:image	Container image.	
spec:containers:image:volumeMounts:mountPath	Mount path of the PersistentVolumeClaim object in the container.	

Parameter	Description	Remarks
spec:volumes:persistentVolumeClaim:claimName	Name of the PersistentVolumeClaim object.	Set this parameter to the name of the PersistentVolumeClaim object created in **4.2 Creating PersistentVolume-Claim4.1 Creating StorageClass**.

Create Pod based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```

4.4 Create Snapshot

The CSI supports snapshot v1beta1 since Kubernetes 1.17. For details, please see: **https://kubernetes-csi.github.io/docs/external-snapshotter.html**

4.4.1 Install Snapshot Beta CRDs per cluster

Before installation these CRDs, you need to confirm whether the snapshot related resource service has been installed. You can execute on the master as follows:

```
kubectl api-resources | grep snapshot | awk'{print $1}'
```

If the following is displayed, there is no need to install again.

```
volumesnapshotclasses
volumesnapshotcontents
volumesnapshots
```

If the result does not include the above services, please follow the instructions below to install it

Step 1 Compile **volumesnapshotclasses.yaml** file if missing volumesnapshotclasses . For details, see sample file **yamls/ snapshot.storage.k8s.io_volumesnapshotclasses.yaml** in the software package.

Create file **volumesnapshotclasses.yaml**.

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

Step 2 Compile **volumesnapshotcontents.yaml** file if missing volumesnapshotcontents . For details, see sample file **yamls/ snapshot.storage.k8s.io_volumesnapshotcontents.yaml** in the software package.

Create file **volumesnapshotcontents.yaml**.

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

Step 3 Compile **volumesnapshots.yaml** file if missing volumesnapshots . For details, see sample file **yamls/snapshot.storage.k8s.io_volumesnapshots.yaml** in the software package.

Create file **volumesnapshots.yaml**.


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

Step 4 Configure according to **section 3.6**

----End

4.4.2 Create VolumeSnapshotClass

Configure the VolumeSnapshotClass yaml file. For details, see sample file **yamls/snapshotclass.yaml** in the software package.

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

Table 4-5 Parameter description

Parameter	Description	Remarks
name	User-defined name of a VolumeSnapshotClass object.	
driver	driver identifier.	The value is fixed to **csi.huawei.com**.
deletionPolicy	The policy of volumeSnapshotContent while VolumeSnapshot is deleted.	Required, optional values are Delete or Retain.

Create VolumeSnapshotClass based on the yaml file. For details, see sample file **yamls/snapshot.yaml** in the software package.

```
# kubectl create -f /path/to/yaml/file
```

4.4.3 Create VolumeSnapshot

Configure the VolumeSnapshot yaml file

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


Table 4-6 Parameter description

Parameter	Description	Remarks
name	User-defined name of a VolumeSnapshot object.	
volumeSnapshotClassName	Name of the VolumeSnapshotClass object.	Set this parameter to the name of the VolumeSnapshotClass object created in **4.4.2-Create VolumeSnapshotClass**
persistentVolumeClaimName	The name of source PersistentVolumeClaim.	Set this parameter to the name of the PersistentVolumeClaim object created in

Create VolumeSnapshot based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```

4.5 Resize

Configure the **StorageClass** yaml file. Add the **allowVolumeExpansion** configuration item. The following is an example:

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

Firstly create StorageClass according to **Section 4.1** and Create PersistentVolumeClaim according to **Section 4.2**. The expansion operation supports online expansion and offline expansion. The operation commands are as follows:

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

"mypvc" is the name of the PVC that needs to be expanded, and **"120Gi"** is the capacity after expansion.

4.6 Create PVC From Source

When creating a new PersistentVolumeClaim, you can create a new PVC through the storage function by specifying the source object. The currently supported sources are PVC and VolumeSnapshot. For details, please click: **https://kubernetes-csi.github.io/docs/volume-datasources.html**.

4.6.1 Clone

Configure the PersistentVolumeClaim yaml file.


```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ***
spec:
  storageClassName: ***
  dataSource:
    name: ***
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: ***Gi
```

Table 4-7 Parameter description

Parameter	Description	Remarks
metadata/ name	User-defined name of a PersistentVolumeClaim object.	
spec/ storageClassName	Name of the StorageClass object.	Set this parameter to the name of the StorageClass object created in **4.1 Creating StorageClass**.
spec/ dataSource/ name	Name of the source PersistentVolumeClaim object	
spec/ resources/ requests/ storage	Size of the volume to be created.	No less than the source PVC capacity. The value format is ***Gi. The unit is GB.

Create PersistentVolumeClaim based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```

4.6.2 Snapshot Restore

Configure the PersistentVolumeClaim yaml file.

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


Table 4-8 Parameter description

Parameter	Description	Remarks
metadata/ name	User-defined name of a PersistentVolumeClaim object.	
spec/ storageClassName	Name of the StorageClass object.	Set this parameter to the name of the StorageClass object created in **4.1 Creating StorageClass**.
spec/ dataSource/ name	Name of the source VolumeSnapshot object	
spec/ resources/ requests/ storage	Size of the volume to be created.	No less than the source VolumeSnapshot capacity. The value format is ***Gi**. The unit is GB.

Create PersistentVolumeClaim based on the yaml file.

```
# kubectl create -f /path/to/yaml/file
```


5 Advanced Features

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

5.1 Creating a Volume in the Specified Storage Backend

In the scenario where multiple storage backends are configured, you can specify a storage backend where you want to create a volume.

Configure the StorageClass yaml file. Add the **backend** configuration item under the **parameters** configuration item. The following is an example:

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

The value of **backend** is the name of a storage backend configured in the **huawei-csi-configmap.yaml** file.

The volume created using the StorageClass object will be created in the specified storage backend.

5.2 Creating a Volume in the Specified Storage Pool

In the scenario where multiple storage pools are configured, you can specify a storage pool where you want to create a volume.

Configure the StorageClass yaml file. Add the **pool** configuration item under the **parameters** configuration item. The following is an example:

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

The value of **pool** is the name of a storage pool.

The volume created using the StorageClass object will be created in the specified storage pool.

5.3 Configuring ALUA

At the scenario of integrating with SAN/Block storage, and multipath involved, ALUA configuration is supported.

5.3.1 Configuring ALUA for Enterprise Storage

Modify the **huawei-csi-configmap.yaml** file. Add the ALUA parameter in the **parameters** configuration item.

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

Table 5-1 ALUA parameter description

Parameter	Description	Remarks
<HostName>	Regular expression of the host name. The worker node with the hostname matching it will use the corresponding ALUA configuration.	If configured as *, it indicates a default configuration. If one worker node cannot match any other <HostName> regular expression, it'll use the default one.
MULTIPATHTYPE	Multipathing type. 0: default 1: uses third-party multipathing software	

Parameter	Description	Remarks
FAILOVERMODE	Initiator switchover mode. 0: early-version ALUA 1: common ALUA 2: ALUA not used 3: special ALUA	This parameter needs to be delivered only when uses third-party multipathing software is enabled, early-version ALUA is not supported in V5 all series.
SPECIALMODETYPE	Initiator special mode type. 0: mode 0 1: mode 1 2: mode 2 3: mode 3	This parameter needs to be delivered only when initiator switchover mode is special ALUA.
PATHTYPE	Initiator path type. 0: optimal path 1: non-optimal path	This parameter needs to be delivered only when uses third-party multipathing software is enabled.

NOTE

1. For different OS, the ALUA configuration may have discrepancy. Visit **https://support.huawei.com/enterprise/zh/index.html**, input "Host Connectivity Guide" in the search bar and start search. At the search results, choose the guide document according with your OS, and configure ALUA according to the recommendation from that guide.
2. This configuration way is suitable for Enterprise V3/V5 storage and Dorado V3.

5.3.2 Configuring ALUA for Dorado V6

Modify the **huawei-csi-configmap.yaml** file. Add the ALUA parameter in the **parameters** configuration item.

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


Table 5-2 ALUA parameter description

Parameter	Description	Remarks
<HostName>	Regular expression of the host name. The worker node with the hostname matching it will use the corresponding ALUA configuration.	If configured as *, it indicates a default configuration. If one worker node cannot match any other <HostName> regular expression, it'll use the default one.
accessMode	Host access mode. 0: "balanced". 1: "asymmetric".	
hyperMetroPathOptimized	Whether the host path to the local HyperMetro array is preferred. 1: Yes 0: No	

NOTE

1. For different OS, the ALUA configuration may have discrepancy. Visit **https://support.huawei.com/enterprise/zh/index.html**, input "Host Connectivity Guide" in the search bar and start search. At the search results, choose the guide document for according with your OS, and configure ALUA according to the recommendation from that guide.
2. This configuration way is suitable for Dorado V6.

5.3.3 Configuring ALUA for Distributed Storage

Modify the **huawei-csi-configmap.yaml** file. Add the ALUA parameter in the **parameters** configuration item.

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


Table 5-3 ALUA parameter description

Parameter	Description	Remarks
<HostName>	Regular expression of the host name. The worker node with the hostname matching it will use the corresponding ALUA configuration.	If configured as *, it indicates a default configuration. If one worker node cannot match any other <HostName> regular expression, it'll use the default one.
switchoverMode	Switchover mode. "Disable_alua": disables ALUA. "Enable_alua": enables ALUA.	
pathType	Path type. "optimal_path": preferred path. "non_optimal_path": non-preferred path.	

NOTE

1. Only available for ISCSI scenario of Distributed Storage.

5.4 Advanced Features of Enterprise Storage

5.4.1 Configuring QoS

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

Configure the **StorageClass** yaml file. Add the **qos** configuration item under the **parameters** configuration item. The following is an example:

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

The value of the **qos** configuration item is JSON character strings in dictionary format. A character string is enclosed by single quotation marks and the dictionary key by double quotation marks.

Volumes created using the StorageClass object has the QoS feature.

Table 5-4 qos parameters

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

NOTE

- **MAXBANDWIDTH** or **MAXIOPS** cannot coexist with **MINBANDWIDTH**, **MINIOPS**, or **LATENCY**.
- For OceanStor Dorado, **IOTYPE** must be set to **2** (read and write I/Os) and **MINBANDWIDTH**, **MINIOPS**, and **LATENCY** are unavailable.

5.4.2 Configuring vStore

Modify the **huawei-csi-configmap.yaml** file and add the **vstoreName** field.

```
{
  "backends": [
  {
    ...
    "user": "***",
    "password": "***",
    "vstoreName": "***"
  }
  ]
}
```

NOTE

user, **password**, and **vstoreName** are the vStore user name, vStore user password, and vStore name configured on the storage device in advance.

5.4.3 Configuring SAN Replication

To use SAN replication, finish the replication relationship configuration between two Huawei storages in advance. Please refer to relevant Huawei storage instruction for specifics.

In the **backends** section in the **huawei-csi-configmap.yaml** file, add two backends that form the replication relationship and add the replicaBackend field for each backend.

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

NOTE

- replicaBackend indicates the peer backend name of two backends forming the replication relationship. As above example shows, replica1's peer backend is replica2, and replica2's peer backend is replica1.

In **parameters** in the **StorageClass** yaml file, add field as below.

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

Table 5-5 Parameter description

Parameter	Description	Remarks
replication	Indicates whether a replication volume is created.	"true" denotes creating a replication volume; Not configuring this parameter or "false" denotes not creating a replication volume.
replicationSyncPeriod	Indicates the interval of replication synchronization	This parameter is optional. Unit is second. Default value is 3600s.

Parameter	Description	Remarks
backend	Indicates the primary backend name where replication volume will be created	Corresponding to the backend name configured in huawei-csi-configmap.yaml.

Volumes created using this StorageClass are volumes with the SAN replication capability.

5.4.4 Configuring SAN HyperMetro

NOTICE

To use SAN HyperMetro, finish the HyperMetro relationship configuration between two Huawei storages in advance. Please refer to relevant Huawei storage instruction for specifics.

In the **backends** section in the **huawei-csi-configmap.yaml** file, add two backends that form the HyperMetro relationship and add the **hyperMetroDomain** field for each backend.

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

NOTE

- **hyperMetroDomain** indicates the HyperMetro domain name configured between Huawei storage systems.
- **metroBackend** indicates the peer backend name of two backends forming the hypermetro relationship. As above example shows, hyperMetro1's peer backend is hyperMetro2, and hyperMetro2's peer backend is hyperMetro1.

In **parameters** in the **StorageClass** yaml file, add field as below.

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


Table 5-6 Parameter description

Parameter	Description	Remarks
hyperMetro	Indicates whether a HyperMetro volume is created.	"true" denotes creating a HyperMetro volume; Not configuring this parameter or "false" denotes not creating a HyperMetro volume.

Volumes created using this StorageClass are volumes with the SAN HyperMetro capability.

5.4.5 Configuring NAS Replication

NOTICE

To use NAS replication, finish the replication relationship configuration between two Huawei storages in advance. Please refer to relevant Huawei storage instruction for specifics.

In the **backends** section in the **huawei-csi-configmap.yaml** file, add two backends that form the replication relationship and add the replicaBackend field for each backend.

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

NOTE

- replicaBackend indicates the peer backend name of two backends forming the replication relationship. As above example shows, replica1's peer backend is replica2, and replica2's peer backend is replica1.
- support vStore. Chapter 5.3.2 describes how to configure vStore.

In **parameters** in the **StorageClass** yaml file, add field as below.

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



```
replicationSyncPeriod: 3600
backend: "***"
```

Table 5-7 Parameter description

Parameter	Description	Remarks
replication	Indicates whether a replication volume is created.	"true" denotes creating a replication volume; Not configuring this parameter or "false" denotes not creating a replication volume.
replicationSyncPeriod	Indicates the interval of replication synchronization	This parameter is optional. Unit is second. Default value is 3600s.
backend	Indicates the primary backend name where replication volume will be created	This parameter is conditional mandatory. At the circumstance of using vStore, if vStore belongs to a replication vStore pair, the replication volume only can be created at the primary end of vStore pair, so it's necessary to specify the primary backend where to create the volume, corresponding to the backend name configured in huawei-csi-configmap.yaml.

Volumes created using this StorageClass are volumes with the NAS replication capability.

5.4.6 Configuring NAS HyperMetro

NOTICE

To use NAS HyperMetro, finish the HyperMetro relationship configuration between two Huawei storages in advance, and create HyperMetro vStore pair. Please refer to relevant Huawei storage instruction for specifics.

In the **backends** section in the **huawei-csi-configmap.yaml** file, add two backends that form the HyperMetro relationship and add the **metrovStorePairID** field for each backend.

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



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

NOTE

- NAS HyperMetro only supports vStore.
- metrovStorePairID indicates the HyperMetro vStore pair ID which vStores belong.
- **metroBackend** indicates the peer backend name of two backends forming the hypermetro relationship. As above example shows, hyperMetro1's peer backend is hyperMetro2, and hyperMetro2's peer backend is hyperMetro1.

In **parameters** in the **StorageClass** yaml file, add field as below.

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

Table 5-8 Parameter description

Parameter	Description	Remarks
hyperMetro	Indicates whether a HyperMetro volume is created.	"true" denotes creating a HyperMetro volume; Not configuring this parameter or "false" denotes not creating a HyperMetro volume.

Volumes created using this StorageClass are volumes with the NAS HyperMetro capability.

5.5 Advanced Features of Distributed Storage

5.5.1 Configuring QoS

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

Configure the **StorageClass** yaml file. Add the **qos** configuration item under the **parameters** configuration item. The following is an example:

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



```
provisioner: "csi.huawei.com"
parameters:
  ...
  qos: '{"maxMBPS": 999, "maxIOPS": 999}'
```

The value of the **qos** configuration item is JSON character strings in dictionary format. A character string is enclosed by single quotation marks and the dictionary key by double quotation marks.

Volumes created using the StorageClass object has the QoS feature.

Table 5-9 qos parameters

Parameter	Description	Remarks
maxMBPS	Maximum bandwidth.	Mandatory. The value is a positive integer, expressed in MB/s.
maxIOPS	Maximum IOPS	Mandatory. The value is a positive integer.

6 FAQ

6.1 Failed to Create Pod Because the iscsi_tcp Service Is Not Started Properly When the Rancher Platform Is Set Up for the First Time

When a user creates Pod, error **Cannot connect ISCSI portal *.*.*.*: libkmod: kmod_module_insert_module: could not find module by name='iscsi_tcp'** is reported in **/var/log/huawei-csi-node**. This is because the iscsi_tcp service may be stopped after the Rancher platform is set up and the iscsi service is installed. You can run the **lsmod | grep iscsi** command to check whether the iscsi_tcp service is started. If the service is not started, run the **modprobe iscsi_tcp** command to start it.

6.2 Failed to Start huawei-csi-node Daemonset, the error message is "/var/lib/iscsi is not a directory

The huawei-csi-node service cannot be started when the huawei-csi-node DaemonSet is started. This is because the container does not have the **/var/lib/iscsi** directory. You just need to **set type whose hostpath is /var/lib/iscsi in the huawei-csi-node.yaml file to "" or delete the type line.**

