eSDK Enterprise Storage Plugins 2.1.RC4

User Guide (Kubernetes CSI)

Issue 01

Date 2019-12-30

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

2.1 System Requirements

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

Table 2-1 System requirements

Item	Requirements
Operating system	• CentOS • Ubuntu
Kubernetes version	1.13/1.14/1.15

2.2 Storage Version Mapping

You can query the version mapping table of eSDK plug-ins to obtain the storage system versions. To obtain the version mapping table, log in to **http://support.huawei.com/enterprise/en/index.html**. In the search box, enter **eSDK Enterprise Storage Plugins** to search for and download the *eSDK Enterprise Storage Plugins x.x.xxx Version Mapping*. *x.x.xxx* indicates the version number.

3 Deployment

3.1 Obtaining the Software Package

You can download the software by visiting **https://support.huawei.com/enterprise/en/software/index.html** and choosing **Cloud Storage > Tools and Platform > eSDK Enterprise Storage Plugins.**The package name is **eSDK_Enterprise_Storage_*.*.**_Kubernetes_CSI_Plugin_*X.X.XX*_*Y*_64.zip**. **.*.*** indicates the release version number and *X.X.XX* indicates the version number. *Y* indicates the running platform, and currently supports x86 and ARM.

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:latest

RUN yum install -y \
    device-mapper-multipath \
    iscsi-initiator-utils \
    xfsprogs \
    nfs-utils \
    net-tools \
    e4fsprogs

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

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

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

```
FROM ubuntu:latest

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

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

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

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.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.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.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.

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

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

----End

3.5 Interconnecting with Cloud Storage

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

3.5.1 Interconnecting with FusionStorage Block 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.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 3-5 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 (-).
storage	String	Type of the storage backend.	The value is fixed to **fusionstorage-sa** 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 Block.
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.	The parameter format is **{"SCSI": {"hostname":"*.*.*.*"}}**, where **hostname** indicates the host name of a worker node and ***.*.*.*** indicates the service IP address of the FusionStorage Block client. If there are multiple worker nodes, configure them in dictionary format.

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

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

----End

3.5.2 Interconnecting with FusionStorage Block 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.

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.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 3-6 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-sa** 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 Block.
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

3.6 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.

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.

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

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**.
volumeType	Type of the volume to be created.	The value is fixed to **lun**.

Parameter	Description	Remarks
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 FS 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.

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.	
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
```


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 Advanced Features of Enterprise Storage

5.3.1 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 5-1 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.

Parameter	Description	Remarks
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.

5.3.2 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.

5.3.3 Configuring HyperMetro

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":[
  {
    ...
    "hyperMetroDomain": "***"
  },
  {
    ...
    "hyperMetroDomain": "***"
  }
  ]
}
```


NOTE

- **hyperMetroDomain** indicates the HyperMetro domain name configured between Huawei storage systems.
- HyperMetro volumes can be created for LUNs, but cannot be created for file systems.

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

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

Table 5-2 Parameter description

Parameter	Description	Remarks
hyperMetro	Indicates whether a HyperMetro volume is created.	

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

